

IMPLEMENTASI METODE ZHANG PADA KALIBRASI KAMERA UNTUK ESTIMASI JARAK PADA OBYEK BERGERAK

Nama Mahasiswa : Shofwan Ali Fauji
NRP : 1212 201 005
Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Penelitian ini membahas mengenai implementasi metode zhang pada kalibrasi kamera stereo. Kamera stereo adalah dua kamera yang disusun sedemikian rupa sehingga dapat digunakan untuk mengamati obyek yang sama dari posisi yang berbeda, sehingga diperlukan kalibrasi kamera untuk mensinkronisasikan kedua kamera yang tersusun dalam sistem kamera tersebut. Penelitian ini mengimplementasikan metode zhang untuk kalibrasi. Proses kalibrasi diawali dengan cara mengatur jarak antara kamera, menggunakan pola planar, menghitung homografi hingga didapatkan parameter-parameter ekstrinsik, intrinsik dan koefisien distorsi dari kedua kamera. Uji coba dilakukan dengan menggunakan parameter-parameter yang telah diperoleh untuk menghilangkan distorsi citra. Parameter terbaik dalam penghilangan distorsi dipilih sebagai parameter untuk estimasi jarak ke suatu obyek yang bergerak. Estimasi jarak terbaik dari hasil uji coba adalah ketika kamera sejajar dengan obyek pada interval jarak 750 sampai 1050 cm dengan median rasio error relatif 0.0648 dan interval jarak 1050 sampai 1350 cm dengan median rasio error relatif 0.0496.

Kata-kunci: Kamera Stereo, Kalibrasi Kamera Zhang, Distorsi Citra, Estimasi Jarak

IMPLEMENTATION OF ZHANG'S METHOD ON CALIBRATION CAMERA FOR ESTIMATING THE DISTANCE OF MOVING OBJECT

Name : Shofwan Ali Fauji
NRP : 1212 201 005
Supervisor : Dr. Budi Setiyono, S.Si, MT

Abstract

This study it is discussed the implementation of zhang's method on camera calibration using stereo cameras. Stereo cameras are two cameras arranged to observe the same object from different positions, therefore , it is need camera calibration to synchronize both cameras arranged in the camera system. In this study, implementing the Zhang's method is used for calibrating. The calibration process is started by adjusting the distance between the camera, using a planar pattern, counting the homography to obtain distortion coefficient, extrinsic and intrinsic parameters and from both of stereo cameras. To test the zhang camera calibration results, experiments are performed to eliminate the image of distortion and selected one of good result to estimate the distance from the camera to a moving object. From the experiments, it can be concluded that the best estimated distance is obtained when the cameras are parallel with the object with the interval 750 to 1050 cm with median relatif error ratio is 0.0648 and the interval 1050 to 1350 cm with median relatif error ratio is 0.0496.

Key-words: *Stereo Camera, Zhang Camera Calibration, Distortion Image, Estimated Distance*

DAFTAR NOTASI

\mathbf{M}	:	Titik tiga dimensi pola planar
\mathbf{m}	:	Titik dua dimensi pola planar
\mathbf{A}	:	Matriks Intrinsik
\mathbf{R}_{kk}	:	Rotasi kamera kanan
\mathbf{R}_{kl}	:	Rotasi kamera kiri
θ	:	Faktor skala dari sumbu x
β	:	Faktor skala dari sumbu y
c	:	Parameter kemiringan
\mathbf{t}	:	Translasi
(u_0, v_0)	:	Koordinat titik pusat
\mathbf{H}	:	Matriks Homografi
k_1, k_2	:	Koefisien distorsi radial
ω	:	Bobot dalam pelatihan GMM
\mathbf{C}	:	Citra RGB
μ_t	:	Vektor nilai mean citra RGB dari gaussian ke k
σ_k	:	Standar deviasi dari gaussian ke k
Σ	:	Matriks kovarian

BAB 2

TINJAUAN PUSTAKA

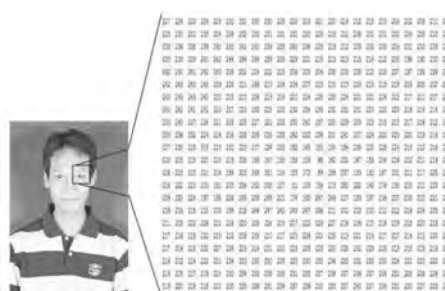
Bab ini membahas mengenai materi-materi yang terkait dalam pengerjaan tesis ini, diantaranya citra, citra video, kamera stereo, kalibrasi kamera, *tracking* obyek, morfologi citra dan distorsi citra. Materi pertama yang dibahas adalah citra pada sub bab 2.1 berikut ini,

2.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu obyek (Sutoyo dkk, 2009). Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra dapat dibagi menjadi dua, yaitu citra analog dan citra digital.

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan dan lain sebagainya. Citra analog tidak dapat dipresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog di antaranya adalah video kamera analog, kamera foto analog dan CT scan.

Citra digital adalah citra yang dapat diolah oleh komputer. Gambar 2.1 menunjukkan citra grayscale berukuran 165×220 piksel diambil sebagian kecil dengan ukuran 22×23 piksel yaitu yang terlihat pada kotak kecil. Jadi sebagian kecil itu adalah contoh bahwa citra digital disimpan memori komputer dalam bentuk angka-angka yang menunjukkan besar intensitasnya pada masing-masing piksel tersebut.



Gambar 2.1: Citra grayscale yang diambil dalam kotak kecil pada gambar itu, disimpan dalam bentuk angka-angka dalam memori komputer

2.2 Citra Video

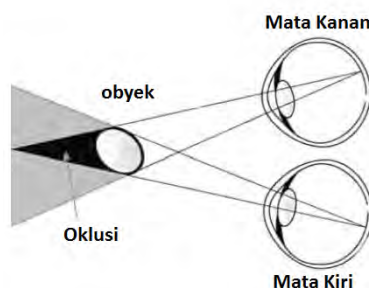
Video adalah teknologi untuk menangkap, merekam, memproses, menyimpan, dan merekonstruksi suatu urutan dari beberapa citra. Video digital merupakan hasil sampling dan kuantisasi dari video analog (Bovik, 2000).

Bagaimanapun juga, video analog yang kita lihat sehari-sehari seperti tampilan pada TV analog, sebenarnya bukan sesuatu yang benar-benar kontinu, melainkan terdiri dari beberapa citra analog yang diputar secara cepat. Sehingga terlihat seperti bergerak. Rangkaian beberapa citra tersebut disebut sebagai *frame* dan kecepatan untuk menggabung citra-citra yang ada dikatakan sebagai *frame rate* dengan *satuan fps (frame per second)*.

Video analog dapat dinyatakan dengan fungsi $I(x, y, t)$, dimana (x, y) adalah nilai kontinu dari fungsi I dan t menyatakan waktu. Sebenarnya tampilan video analog di TV maupun monitor merupakan representasi dari fungsi sinyal elektrik satu dimensi $V(t)$. Dimana sinyal elektrik satu dimensi tersebut terdiri dari beberapa citra analog $I(x, y, t)$ dengan jumlah citra (x, y) tertentu dan waktu (t) tertentu. Proses pemisahan video ke beberapa unit frame citra disebut sebagai scanning.

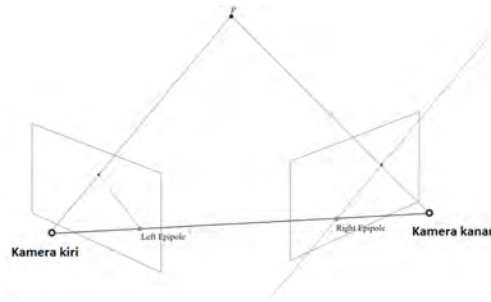
2.3 Kamera Stereo

Kamera stereo adalah dua kamera yang mengamati obyek yang sama dari posisi yang berbeda (Hanning, 2011). Ide dasar dari kamera stereo adalah dua mata manusia yang selalu melihat obyek yang sama, meskipun berbeda sudut pandang seperti pada Gambar 2.2. Cara yang sederhana untuk mengkalibrasi kamera stereo yaitu dengan mengkalibrasi setiap kamera secara terpisah. Perbedaan yang fundamental bila menggunakan satu citra dengan menggunakan dua citra dari kamera stereo, yaitu dua citra atau lebih yang dihasilkan sistem kamera stereo bisa didapatkan informasi yang lebih dalam. Informasi yang lebih dalam tersebut diantaranya geometri, informasi 3 dimensi (3D) bisa dipulihkan melalui proses *triangulation* (Cyganek, Siebert, 2009).



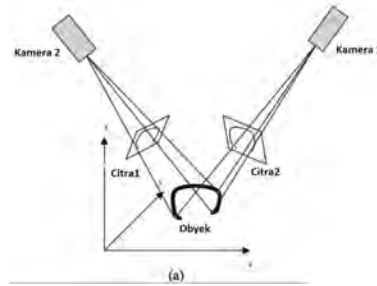
Gambar 2.2: Sistem mata manusia

Terinspirasi dari sistem mata manusia seperti pada Gambar 2.2 di atas, kemudian disusun sistem kamera stereo seperti pada Gambar 2.3.



Gambar 2.3: Sistem kamera stereo

Sistem pada Gambar 2.3, disimulasikan seperti pada Gambar 2.4.



Gambar 2.4: Simulasi sistem kamera stereo

2.4 Kalibrasi Kamera

Kalibrasi kamera bertujuan mensinkronisasikan dua kamera stereo. Kedua bola mata manusia contohnya, saat kedua bola mata manusia dipusatkan pada obyek tertentu, maka akan ada informasi yang bisa didapatkan seperti estimasi jarak dan 3 dimensi dari obyek tersebut. Serangkaian proses kalibrasi kamera berfungsi untuk mendapatkan parameter transformasi antara lensa kamera yang digunakan dalam mengambil citra pada obyek (Wöhler, 2009). Parameter yang dihasilkan berupa parameter ekstrinsik dan intrinsik. Parameter ekstrinsik meliputi translasi dan rotasi, sedangkan parameter intrinsik meliputi koordinat titik pusat citra, faktor skala dan parameter kemiringan.

2.4.1 Kalibrasi Kamera Zhang

Kalibrasi kamera sangat dibutuhkan dalam komputer visi 3D dengan menggali informasi dari citra 2D. Titik 2D dinotasikan dengan $\mathbf{m} = [\mathbf{u}, \mathbf{v}]^T$. Titik 3D dinotasikan dengan $\mathbf{M} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^T$ dengan \mathbf{z} adalah vektor nol (Zhang, 1999). Kalibrasi kamera menggunakan dua kamera yang disusun menjadi kamera stereo, sehingga dari dua kamera stereo ini juga didapatkan dua citra yang memiliki informasi yang sebagian besar sama. Oleh karena itu perlu dihitung relasi koordinat piksel dari dua citra atau yang disebut homografi. Matriks homografi dihitung dengan cara mencari terlebih dahulu korespondensi 2D dan 3D dengan persamaan (2.1) berikut,

$$\mathbf{X} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_i x_i & -v_i y_i \end{bmatrix} \quad (2.1)$$

untuk $i = 1, 2, 3, \dots, n$, dimana n dalam persamaan ini adalah banyaknya titik pola planar yang terdeteksi. Langkah selanjutnya menyelesaikan persamaan linier $\mathbf{X} \mathbf{tvec} = \mathbf{U}$ untuk mendapatkan vektor \mathbf{tvec} , dimana $\mathbf{U} = [u^T \ v^T]$. Apabila persamaan $\mathbf{X} \mathbf{tvec} = \mathbf{U}$ tidak bisa diselesaikan, maka vektor \mathbf{tvec} tidak bisa didapatkan dan kalibrasi tidak bisa dilakukan, sehingga harus mengganti data yang digunakan dengan data yang lain. Vektor \mathbf{tvec} yang dihasilkan akan berbentuk seperti pada persamaan (2.2) berikut,

$$\mathbf{tvec} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{bmatrix} \quad (2.2)$$

Langkah selanjutnya menghitung matriks \mathbf{T} menggunakan persamaan (2.3) berikut ini,

$$\mathbf{T} = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \\ P_7 & P_8 & P_9 \end{bmatrix}^{-1} \quad (2.3)$$

P_9 pada matriks homografi bernilai satu. Setiap elemen pada matriks \mathbf{T} dibagi dengan $\mathbf{T}(3,3)$. Matriks \mathbf{T} yang setiap elemennya telah dibagi dengan $\mathbf{T}(3,3)$, ditranpose untuk mendapatkan matriks homografi yang ditulis pada persamaan (2.4) berikut ini,

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix} \quad (2.4)$$

Matriks homografi pada setiap kamera yang didapatkan, digunakan untuk menyusun matriks $\mathbf{V} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]$. Langkah selanjutnya adalah mendekomposisi matriks \mathbf{V} menggunakan metode *singular value decomposition* didapatkan matriks \mathbf{D} yang merupakan singular vektor kanan dari \mathbf{V} . Persamaan (2.5) berikut adalah bentuk matriks \mathbf{D} yang akan dihasilkan,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}(1,1) & \mathbf{D}(1,2) & \mathbf{D}(1,3) & \cdots & \mathbf{D}(k,1) \\ \mathbf{D}(2,1) & \mathbf{D}(2,2) & \mathbf{D}(2,3) & \cdots & \mathbf{D}(k,2) \\ \mathbf{D}(3,1) & \mathbf{D}(3,2) & \mathbf{D}(3,3) & \cdots & \mathbf{D}(k,3) \\ \mathbf{D}(4,1) & \mathbf{D}(4,2) & \mathbf{D}(4,3) & \cdots & \mathbf{D}(k,4) \\ \mathbf{D}(5,1) & \mathbf{D}(5,2) & \mathbf{D}(5,3) & \cdots & \mathbf{D}(k,5) \\ \mathbf{D}(6,1) & \mathbf{D}(6,2) & \mathbf{D}(6,3) & \cdots & \mathbf{D}(k,6) \end{bmatrix} \quad (2.5)$$

k adalah panjang kolom matriks \mathbf{D} . Sehingga dengan menggunakan matriks kolom terakhir dari matriks \mathbf{D} dapat disusun matriks \mathbf{B} seperti persamaan (2.7) berikut ini,

$$\mathbf{B} = \begin{bmatrix} \mathbf{D}(k,1) & \mathbf{D}(k,2) & \mathbf{D}(k,4) \\ \mathbf{D}(k,2) & \mathbf{D}(k,3) & \mathbf{D}(k,5) \\ \mathbf{D}(k,4) & \mathbf{D}(k,5) & \mathbf{D}(k,6) \end{bmatrix} \quad (2.6)$$

$$= \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \quad (2.7)$$

Matriks \mathbf{B} yang didapatkan digunakan mengestimasi parameter intrinsik pada kamera stereo menggunakan persamaan (2.8-2.13) berikut ini,

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{13}B_{22} - B_{12}^2} \quad (2.8)$$

$$\lambda = B_{33} - \frac{B_{12}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \quad (2.9)$$

$$\theta = \sqrt{\frac{\lambda}{B_{11}}} \quad (2.10)$$

$$\beta = \sqrt{\frac{\lambda B_{11}}{(B_{11}B_{22} - B_{12}^2)}} \quad (2.11)$$

$$\gamma = \frac{-B_{12}\theta^2\beta}{\lambda} \quad (2.12)$$

$$u_0 = \frac{\lambda v_0}{\beta} - \frac{B_{13}\theta^2}{\lambda} \quad (2.13)$$

Parameter-parameter intrinsik yang sudah didapatkan disusun menjadi matriks \mathbf{A} seperti pada persamaan (2.14). Penyusunan parameter-parameter intrinsik menjadi matriks \mathbf{A} adalah untuk menghitung parameter ekstrinsik.

$$\mathbf{A} = \begin{bmatrix} \theta & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Tiga vektor kolom dari matriks homografi, dimisalkan dengan $\mathbf{h}_1, \mathbf{h}_2$ dan \mathbf{h}_3 . Tiga vektor kolom tersebut digunakan untuk menghitung parameter ekstrinsik, tapi terlebih dahulu dihitung $\Lambda_{r1}, \Lambda_{r2}$ dan Λ_{r3} pada persamaan (2.15-2.17)

$$\Lambda_{r1} = \frac{1}{\|\mathbf{A}^{-1}\mathbf{h}_1\|} \quad (2.15)$$

$$\Lambda_{r2} = \frac{1}{\|\mathbf{A}^{-1}\mathbf{h}_2\|} \quad (2.16)$$

$$\Lambda_{r3} = \frac{\Lambda_{r1} + \lambda \mathbf{r}_2}{2} \quad (2.17)$$

Nilai $\Lambda_{r1}, \Lambda_{r2}$ dan Λ_{r3} yang didapatkan, digunakan untuk menghitung matriks rotasi dan translasi. Matriks rotasi dibangun dari nilai $\mathbf{r}_1, \mathbf{r}_2$ dan \mathbf{r}_3 yang disusun seperti pada persamaan (2.22). Sedangkan untuk menghitung parameter translasi digunakan persamaan (2.21).

$$\mathbf{r}_1 = \Lambda_{r1}\mathbf{A}^{-1}\mathbf{h}_1 \quad (2.18)$$

$$\mathbf{r}_2 = \Lambda_{r2}\mathbf{A}^{-1}\mathbf{h}_2 \quad (2.19)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (2.20)$$

$$\mathbf{t} = \Lambda_{r3}\mathbf{A}^{-1}\mathbf{h}_3 \quad (2.21)$$

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \quad (2.22)$$

Hasil kalibrasi kamera zhang akan diimplementasikan untuk estimasi jarak obyek yang bergerak. Untuk mendeteksi jarak obyek bergerak, ada satu langkah yang harus dilakukan sebelum menghitung jarak, yaitu deteksi obyek bergerak. Deteksi obyek bergerak pada umumnya disebut *tracking* obyek. Subbab selanjutnya akan dibahas mengenai *tracking* obyek beserta metode yang digunakan seperti *background subtraction* dan *gaussian mixture model*.

2.5 Tracking Obyek

Dua hal yang perlu diperhatikan pada estimasi jarak pada obyek yang bergerak adalah deteksi obyek bergerak dan pemisahan *background* dengan *foreground*. Metode yang digunakan dalam penelitian ini untuk mendapatkan obyek yang bergerak adalah metode *background subtraction*. Sedangkan pada pemisahan *background* dan *foreground* digunakan metode GMM, karena metode ini dapat digunakan dalam berbagai kondisi citra.

Tracking obyek bagian dari deteksi obyek. Deteksi obyek melakukan pendeteksian pada obyek yang diam sedangkan *tracking* obyek mendeteksi obyek yang bergerak. *Tracking* obyek menggunakan sensor untuk melakukan pengukuran, sehingga dapat menentukan lokasi, jalan dan karakteristik obyek yang menarik (Challa, Morelande, Mušicki, 2011). *Tracking* obyek pada penelitian ini menggunakan algoritma *background subtraction* berbasis *gaussian mixture model*. Berikut ini pembahasan mengenai *background subtraction* dan *gaussian mixture model*.

2.5.1 Algoritma *Background Subtraction*

Background subtraction (BS) digunakan untuk mendeteksi *foreground* atau obyek yang bergerak pada rangkaian frame dalam citra video. Setiap frame citra dari video dikurangi dengan model background yang telah dipilih $\mathbf{B}(x, y, t)$ seperti pada persamaan (2.23). Apabila nilai citra hasil pengurangan memenuhi ambang batas seperti pada persamaan (2.24) maka nilai citra tersebut terdeteksi sebagai obyek atau *foreground* $\mathbf{f}(x, y, t)$.

$$|\mathbf{I}(x, y, t) - \mathbf{B}(x, y, t)| = \mathbf{d}(x, y, t) \quad (2.23)$$

$\mathbf{I}(x, y, t)$ adalah citra pada frame ke- t

$$\mathbf{f}(x, y, t) = \begin{cases} 1 & \text{Jika } d(x, y, t) \geq \tau \\ 0 & \text{Jika selain itu} \end{cases} \quad (2.24)$$

τ adalah nilai ambang batas.

2.5.2 Metode *Gaussian Mixture Model*

Gaussian mixture model (GMM) adalah metode yang bisa digunakan dalam *background* citra yang selalu stasis, multimodal, maupun yang mengandung *noise* (Benezith, dkk, 2012).

Jumlah model *background* GMM yang digunakan mempengaruhi waktu komputasi dalam deteksi obyek, tapi semakin besar jumlah model *background* yang ditetapkan maka ada kemungkinan untuk menghasilkan deteksi yang lebih baik. Suatu citra sebelum masuk dalam algoritma GMM, harus dimorfologi terlebih dahulu untuk mempermudah dalam pemisahan

background dan *foreground*. Morfologi citra akan dibahas lebih detail pada sub-bab selanjutnya, oleh karena itu berikut ini ditampilkan terlebih dahulu

Secara singkat, terdapat tiga proses pada GMM:

1. Tahap inisialisasi dan Seleksi Input

Langkah pertama dalam perancangan GMM adalah memberikan nilai awal varian (σ^2), banyaknya pelatihan frame, laju pelatihan (α), banyaknya model *background*. Pertama-tama menghitung probabilitas piksel \mathbf{C}^n vektor dari citra RGB pada waktu ke- n menggunakan persamaan (4.15) berikut ini (Stauffer, Grimson, 1999),

$$p(\mathbf{C}^n) = \sum_{j=1}^k \omega_k \eta(\mathbf{C}^n | \theta_j) \quad (2.25)$$

dimana ω_k adalah bobot dari komponen gaussian ke- k . η adalah distribusi normal komponen ke- k yang dihitung menggunakan persamaan berikut ini,

$$\eta(\mathbf{C} | \theta_j) = \eta(\mathbf{C} | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{C} - \mu_k)^T \Sigma_k^{-1} (\mathbf{C} - \mu_k)}$$

μ_k adalah mean dan $\Sigma_k = \sigma^2 \mathbf{I}$ yang merupakan kovarian ke- k . $|\Sigma|$ adalah determinan dari kovarian. Distribusi pada persamaan (4.16) adalah distribusi untuk menentukan update parameter, apabila tidak terpenuhi maka M^{n+1} bernilai nol dan parameter yang diperbarui hanya parameter bobot. Apabila terpenuhi maka M^{n+1} bernilai satu dan parameter yang diperbarui adalah bobot, mean dan kovarian.

$$\mu_t - 2.5 * \sigma_k < \mathbf{C}^n < \mu_t + 2.5 * \sigma_k \quad (2.26)$$

2. Tahap Memperbarui Parameter

Tahap pertama telah menghasilkan parameter apa saja yang diperbarui. Persamaan yang digunakan untuk memperbarui nilai dari ω_k^{n+1} (bobot), μ^{n+1} (mean) dan kovarian Σ_k ditampilkan pada persamaan-persamaan berikut ini (Kaewtrakulpong, Bowden, 2001),

$$\omega_k^{n+1} = (1 - \alpha) \omega_k^n + \alpha (M^{n+1}) \quad (2.27)$$

$$\Sigma_k^{n+1} = (1 - \alpha)\Sigma_k^n + \rho(\mathbf{C}^{n+1} - \mu_k^{n+1})(\mathbf{C}^{n+1} - \mu_k^{n+1})^T \quad (2.28)$$

$$\mu_k^{n+1} = (1 - \alpha)\mu_k^n + \rho\mathbf{C}^{n+1}$$

$$\rho = \alpha\eta(\mathbf{C}^{n+1} | \mu_k^n, \Sigma_k^n)$$

3. Tahap pemilihan distribusi yang mencerminkan *background*

Langkah selanjutnya adalah memilih model yang menyerupai *background*. Setiap piksel dalam *schene* dimodelkan dengan K distribusi gaussian. Model akan diurutkan berdasarkan $\frac{\omega}{\sigma^2}$, dengan distribusi yang paling menyerupai *background* (B) berada di bagian atas. Persamaan untuk memilih B distribusi pertama yang dijadikan *background* adalah,

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k < L \right) \quad (2.29)$$

Threshold L adalah nilai minimum yang harus dipenuhi oleh model *background*.

2.6 Morfologi Citra

Morfologi citra telah sedikit disinggung peran pentingnya pada sub-bab *tracking* obyek. Morfologi citra sendiri dalam pengolahan citra digital digunakan untuk mengolah struktur suatu obyek dalam citra sehingga diperoleh struktur obyek yang diinginkan (Sutoyo dkk, 2009). Berikut ini dibahas mengenai empat morfologi citra yaitu filtering, dilasi, erosi dan filling.

2.6.1 Filtering

Filtering berfungsi untuk membiarkan komponen frekuensi tertentu dan menghilangkan komponen frekuensi tertentu. Filtering yang digunakan di sini adalah filter Gaussian. Setiap piksel pada filter Gaussian diganti dengan rata-rata dari nilai pembobotan untuk setiap piksel-piksel tetangganya dan piksel itu sendiri (Sutoyo dkk, 2009). Filter Gaussian dirancang persamaan (2.30) berikut ini,

$$g(x, y) = c.e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.30)$$

c adalah konstanta normalisasi. Nilai dari σ^2 adalah nilai yang bisa ditentukan untuk mendapatkan nilai dari c . Contoh, apabila dibuat filter gaussian dengan ukuran (3×3) , dan dirancang nilai dari $g(1, 1) = 1$, kemudian ditentukan bahwa nilai dari $\sigma^2 = 2$. Matriks yang dihasilkan menggunakan persamaan $e^{-\frac{x^2+y^2}{2\sigma^2}}$ dapat dilihat pada Tabel 2.1 berikut ini,

Tabel 2.1: Perancangan filter gaussian

(x, y)	-1	0	1
-1	0.7788	0.8825	0.7788
0	0.8825	1	0.8825
1	0.7788	0.8825	0.7788

Nilai $g(1, 1)$ sudah ditentukan, sehingga nilai c dapat dihitung sebagai berikut,

$$g(1, 1) = c.e^{-\frac{3^2+3^2}{2 \cdot 2^2}}$$

$$1.2840 = c$$

Nilai c yang sudah didapatkan digunakan untuk menghitung kembali matriks pada tabel Tabel 2.1 menggunakan persamaan (2.30), hasil perhitungan yang telah dibulatkan ditampilkan pada Tabel 2.2 berikut ini,

Tabel 2.2: Perhitungan filter gaussian

$g(x, y)$	-1	0	1
-1	1	1	1
0	1	1	1
1	1	1	1

Nilai piksel hasil perhitungan harus dinormalkan dengan cara membaginya dengan jumlah semua elemen nilai nilai piksel, matriks tersebut memiliki jumlah semua elemen 9. Rancangan filter gaussian menjadi seperti berikut ini,

$$g(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Setiap piksel pada citra yang difiltering, akan diganti dengan rata-rata dari nilai pembobotan untuk setiap piksel-piksel tetangganya. Contohnya dimiliki citra dengan matriks,

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Nilai dari 3 akan diganti dengan pembulatan rata-rata dari nilai pembobotan untuk setiap piksel-piksel tetangganya, nilai pembulatan rata-rata dari nilai pembobotan untuk setiap piksel-piksel tetangganya adalah $\frac{1}{9}[(1*1)+(1*2)+(1*1)+(1*1)+(1*2)+(1*1)+(1*1)+(1*2)+(1*1)] = 1$.

2.6.2 Dilasi dan Erosi

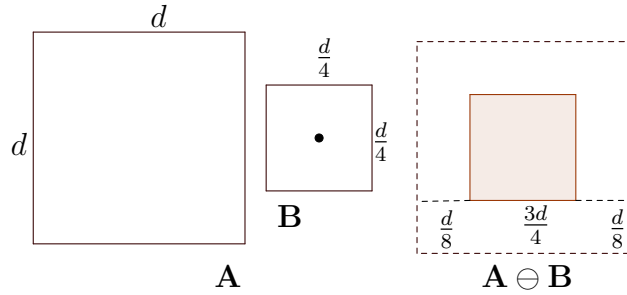
Tujuan dari erosi adalah untuk memperkecil noise-noise yang ada agar hilang, kemudian komponen frekuensi yang tersisa diperbesar kembali dengan dilasi. Erosi dari citra **A** oleh **B** adalah himpunan seluruh piksel x sedemikian rupa sehingga **B** ditranslasi oleh x , yang berada dalam citra **A**. Erosi **A** oleh **B** dinotasikan \ominus , secara matematis dapat ditulis sebagai berikut (Sutoyo dkk, 2009),

$$\mathbf{A} \ominus \mathbf{B} = \{x | (\mathbf{B})_x \subseteq \mathbf{A}\} \quad (2.31)$$

Sedangkan dilasi **A** oleh **B** dinotasikan \oplus , secara matematis dapat ditulis sebagai berikut,

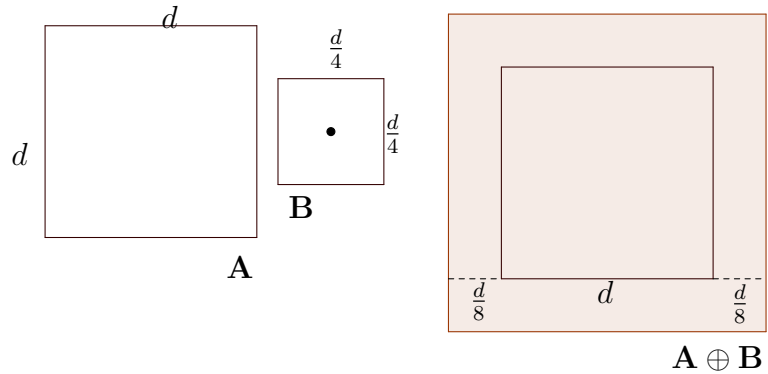
$$\mathbf{A} \oplus \mathbf{B} = \{x | [(\mathbf{B})_x \cap \mathbf{A}] \neq \emptyset\} \quad (2.32)$$

Contoh erosi ditunjukkan pada Gambar 2.5 berikut ini,



Gambar 2.5: Contoh erosi dari citra **A** oleh **B**

Contoh dilasi ditunjukkan pada Gambar 2.6 berikut ini,



Gambar 2.6: Contoh dilasi dari citra **A** oleh **B**

2.6.3 Filling

Filling berguna untuk mengisi bagian-bagian yang masih dalam batas dengan nilai satu, dan bagian di luar batas dengan nilai nol. Batas filling (ψ) dihitung dengan cara mengurangi citra dengan hasil erosi citra atau bisa ditulis seperti persamaan (2.33) berikut,

$$\psi(\mathbf{A}) = \mathbf{A} - (\mathbf{A} \ominus \mathbf{B}) \quad (2.33)$$

2.7 Distorsi Citra

Distorsi adalah penyimpangan bujur sangkar. Kondisi citra yang masih terdistorsi dapat mempengaruhi estimasi jarak dari obyek ke kamera. Oleh karena itu, salah satu proses yang sangat penting adalah menghilangkan distorsi citra agar penyimpangan pada bujur sangkar dapat diperbaiki. Banyak sekali jenis distorsi yang ada, tapi ada dua yang umumnya digunakan, yaitu distorsi radial dan *vignetting* (Montabone, 2010). Penelitian ini menggunakan distorsi radial, sehingga dalam pelaksanaannya dilakukan estimasi koefisien distorsi radial dari citra. Koefisien tersebut yang digunakan untuk memperbaiki distorsi pada citra. Ada dua tipe distorsi radial, yaitu distorsi *barrel* and *pincushion*. Distorsi *barrel* adalah pembengkokan ke arah depan pada pusat citra. Contoh dari distorsi *barrel* ditunjukkan pada Gambar 2.7 berikut ini,



Gambar 2.7: Distorsi *barrel* pada citra

Distorsi *pincushion* adalah kebalikan dari distorsi *barrel*, karena pada

distorsi ini terjadi pembengkokan ke dalam pada pusat citra. Contoh dari distorsi *pincushion* ditunjukkan pada Gambar 2.8 berikut ini,



Gambar 2.8: Distorsi *pincushion* pada citra

Gambar 2.7 koefisien distorsi radialnya adalah -37.662 . Apabila distorsinya dihilangkan maka pembengkokan di pusat citra dapat diperbaiki. Gambar 2.9 merupakan Gambar 2.7 yang telah dihilangkan distorsinya,



Gambar 2.9: Citra setelah dihilangkan distorsi *barrel*-nya

BAB 3

METODE PENELITIAN

Bagian ini menguraikan beberapa metode penelitian yang akan digunakan atau dikerjakan untuk mencapai tujuan penelitian.

3.1 Tahapan Penelitian

1. Studi literatur
Studi literatur mengenai kalibrasi kamera dan deteksi obyek bergerak.
2. Merangkai sistem kamera stereo
Merangkai dua web cam yang akan digunakan sesuai dengan sistem kamera stereo. Kemudian mengintegrasikannya dengan PC.
3. Kalibrasi kamera zhang
Melakukan kalibrasi kamera zhang menggunakan pola planar yang telah dijadikan citra dengan kamera stereo.
4. Uji Coba Kombinasi Pola Planar pada Kalibrasi Kamera
Uji Coba Kombinasi Pola Planar dilakukan untuk mendapatkan hasil terbaik dalam kalibrasi kamera saat dilakukan uji coba. Kombinasi pola planar meliputi posisi pola planar dan jumlah pola planar yang digunakan.
5. Uji Coba Hasil Kalibrasi Kamera Zhang pada Penghilangan Distorsi Citra
Hasil-hasil kalibrasi pola planar yang telah didapatkan, diuji coba dengan menghilangkan distorsi pada citra, untuk mengetahui hasil kombinasi terbaik.
6. Uji Coba Hasil Kalibrasi Kamera pada Estimasi Jarak pada Obyek Bergerak
Kombinasi terbaik yang telah didapatkan pada pola planar dalam melakukan kalibrasi kamera. Diuji coba lagi untuk yang kedua kali, dengan cara melakukan estimasi jarak pada obyek bergerak. Sehingga setiap frame akan dihilangkan distorsinya dan dideteksi obyek yang bergerak, kemudian jarak obyek yang bergerak tersebut diestimasi.
7. Menganalisis hasil dan pembahasan
Analisis hasil dilakukan setelah melihat hasil uji coba parameter-parameter yang dihasilkan dari kalibrasi kamera zhang. Hasil simulasi tersebut merupakan kesimpulan dari pembahasan ini.

8. Menyusun Laporan

Penulisan tesis, makalah dan diseminasi. Format yang digunakan dalam penulisan tesis ini adalah format yang telah dipublikasikan oleh pascasarjana ITS. Penulisan makalah disesuaikan dengan format yang diatur oleh pihak panitia yang mengadakan seminar nasional yang akan diikuti.

BAB 4

ANALISIS SISTEM DAN PERANCANGAN PERANGKAT LUNAK

Bab ini akan menjelaskan mengenai analisis sistem, perancangan kamera stereo, pengolahan data dan kalibrasi kamera zhang beserta uji cobanya pada penghilangan distorsi citra dan estimasi jarak obyek bergerak. Hal ini bertujuan untuk memberikan gambaran tentang bagaimana cara kerja kalibrasi kamera dan bagaimana implementasi pada perangkat lunak yang akan dibuat. Sehingga didapatkan proses terbaik dalam melakukan kalibrasi kamera zhang.

4.1 Analisis Sistem

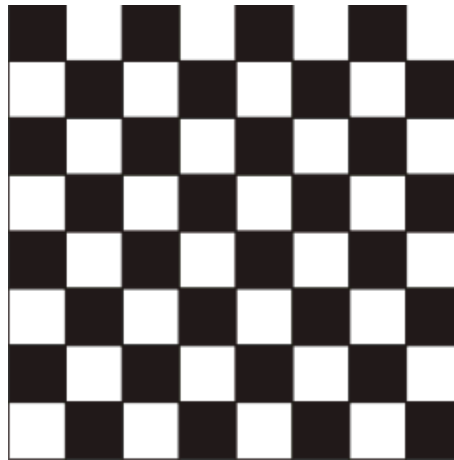
Tujuan utama dari analisis sistem adalah memberikan gambaran keseluruhan mengenai proses kerja perangkat lunak yang akan dibuat dan bagaimana sistem melakukan kalibrasi kamera hingga mendapatkan estimasi jarak obyek bergerak. Perangkat lunak yang dihasilkan, secara berurutan melakukan kalibrasi kamera kemudian menghilangkan distorsi citra dan estimasi jarak setiap obyek yang bergerak, sehingga bisa didapatkan proses terbaik dalam kalibrasi kamera.

Proses pertama yang dilakukan adalah dengan menyusun kamera stereo. Kamera stereo yang digunakan adalah dua buah kamera logitech c170 yang disusun menjadi kamera stereo. Kamera stereo disusun menyerupai sistem mata manusia, oleh karena itu dua buah kamera logitech c170 disusun seperti pada Gambar 4.2 dengan jarak 12 cm. Sehingga didapatkan dua kamera dengan sudut pandang yang berbeda untuk menangkap obyek yang sama.

Jarak 12 cm dipilih karena dari jarak tersebut sudah didapatkan orientasi yang berbeda tapi dengan ruang informasi bersama dari dua kamera tersebut yang masih besar. Dua kamera stereo tersebut disinkronisasikan dengan kalibrasi kamera, sehingga didapatkan parameter ekstrinsik, intrinsik beserta koefisien distorsi. Parameter ekstrinsik dan intrinsiknya yang didapatkan perlu diuji coba untuk mengetahui kualitas parameter-parameter tersebut. Uji coba parameter-parameter yang didapatkan dari hasil kalibrasi kamera meliputi menghilangkan distorsi citra dan estimasi jarak obyek bergerak.

Kalibrasi kamera yang digunakan adalah kalibrasi kamera zhang dengan mengolah pola planar yang telah dicetak pada sebuah papan. Pola planar digunakan karena pola ini mudah untuk melakukan korespondensi antara dua citra. Pola planar adalah pola kotak-kotak dengan warna kotak hitam dan putih pada kotak tetangga samping, atas maupun bawahnya, dan begitu juga sebaliknya. Pola planar juga digunakan pada pola papan catur. Pola planar

dapat dicontohkan pada Gambar 4.1.



Gambar 4.1: Pola planar

4.2 Perancangan Kamera Stereo

Dua web cam yang digunakan, akan disusun sesuai dengan sistem kamera stereo. Kamera disusun dengan jarak 12 cm, dengan jarak ini citra dari kamera kanan akan sedikit bergeser pada citra dari kamera kiri sehingga ruang informasi bersama yang ingin didapatkan dari dua kamera bisa lebih besar. Gambar kamera stereo ditunjukkan pada Gambar 4.2 berikut ini,



Gambar 4.2: Kamera stereo yang digunakan

Kamera stereo inilah yang digunakan untuk melakukan pengambilan data, baik data pola planar untuk kalibrasi kamera maupun data uji coba parameter-parameter hasil kalibrasi kamera.

4.2.1 Pengambilan Data

Data yang digunakan untuk kalibrasi kamera adalah pola planar yang dicetak pada sebuah papan. Langkah selanjutnya, pola itu diambil citra digitalnya menggunakan kamera kiri dan kanan yang telah tersusun menjadi kamera stereo, seperti pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.3: Pola planar pada kamera kiri



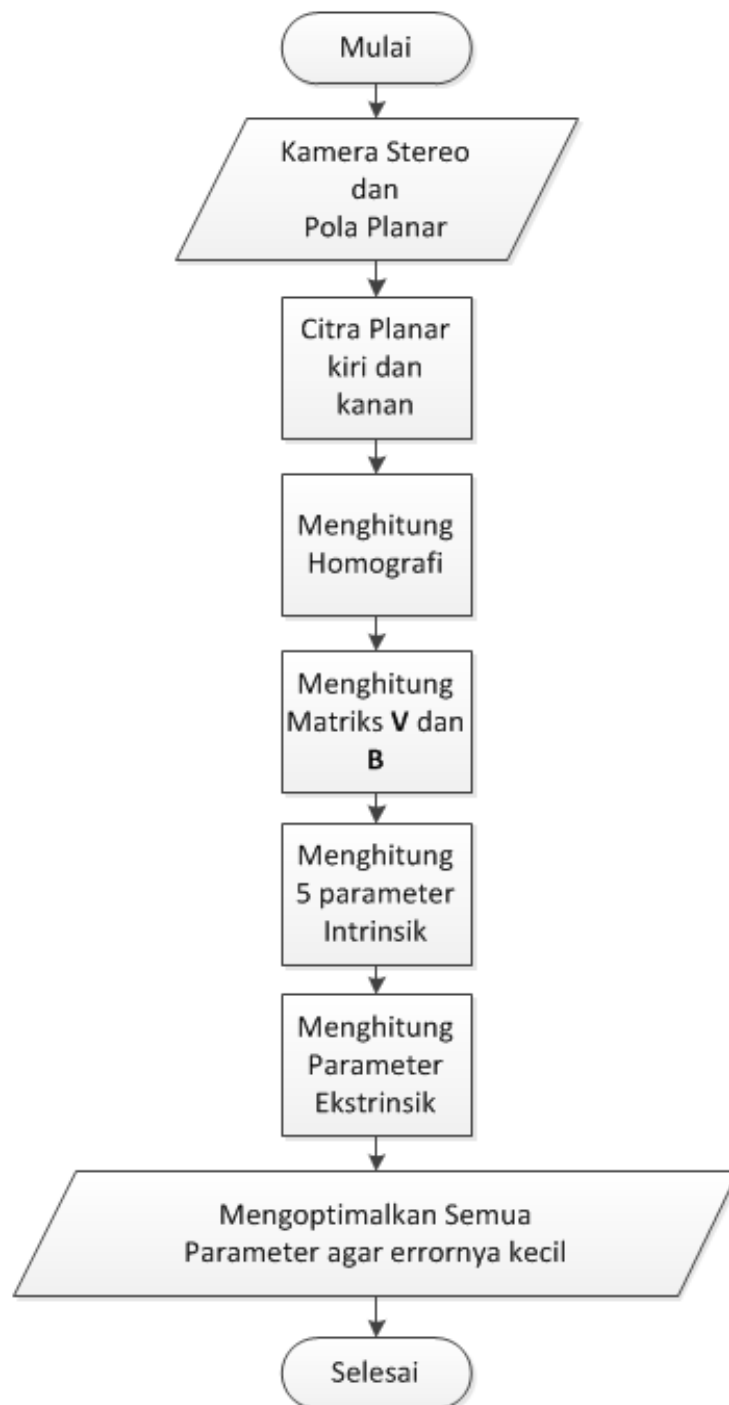
Gambar 4.4: Pola planar pada kamera kanan

4.3 Pengolahan Data

Pola planar yang didapatkan baik dari kamera kiri maupun kanan, bisa digunakan untuk melakukan kalibrasi kamera menggunakan metode zhang. Secara singkat prosedur kalibrasi kamera zhang dapat di rangkum dalam langkah-langkah berikut,

- 1 Mencetak pola planar dan menempatkannya pada tempat yang akan di ambil citra-nya.
- 2 Mengambil beberapa citra dari obyek dengan sudut pandang yang berbeda. Bisa dengan memindahkan kamera atau pola planar yang dipindahkan.
- 3 Mendeteksi setiap titik yang ada pada pola planar.
- 4 Mengestimasi parameter instrinsik dan ekstrinsik.
- 5 Mengoptimalkan semua parameter agar errornya kecil.

Gambar 4.5 ini merupakan flowcart kalibrasi kamera,



Gambar 4.5: Flowcart kalibrasi kamera

4.3.1 Estimasi Parameter Ekstrinsik dan Intrinsik Menggunakan Kalibrasi Kamera Zhang

Berikut langkah-langkah estimasi parameter ekstrinsik dan intrinsik menggunakan kalibrasi kamera zhang,

1. Setiap pola planar baik yang dihasilkan dari kamera kiri maupun kanan dideteksi setiap titik pada pola planarnya. Setiap koordinat titik pola planar yang terdeteksi dimisalkan (\mathbf{u}, \mathbf{v}) (Zhang, 1999).
2. Menentukan koordinat titik dunia, dengan syarat diketahui ukuran sisi setiap persegi dalam pola planar. Tujuannya untuk mengetahui posisi titik pola planar pada dunia nyata, dengan titik ujung kiri koordinatnya $(0,0)$. Koordinat titik dunia dimasukkan dalam variabel (\mathbf{x}, \mathbf{y}) . Koordinat titik pada pola planar baik dalam citra maupun titik dunia akan digunakan untuk mengestimasi parameter intrinsik.
3. Menghitung homografi (\mathbf{H}) pada setiap posisi kamera.

Homografi adalah relasi koordinat piksel dari dua citra. Menghitung matriks homografi citra membutuhkan koordinat titik dunia (\mathbf{x}, \mathbf{y}) dan koordinat titik pola planar (\mathbf{u}, \mathbf{v}) yang disusun seperti pada persamaan (2.1). Persamaan (2.1) dapat dijabarkan menjadi (4.1) berikut ini,

$$\mathbf{X} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -u_nx_n & -u_ny_n \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2x_2 & -v_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_n & y_n & 1 & -v_nx_n & -v_ny_n \end{bmatrix} \quad (4.1)$$

Tranpose matriks \mathbf{u} dan \mathbf{v} disusun seperti pada persamaan (4.2) untuk menghasilkan matriks \mathbf{U} .

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \end{bmatrix} \quad (4.2)$$

Vektor kolom \mathbf{U} berfungsi untuk menghitung vektor kolom \mathbf{tvec} yang merupakan pembentuk matriks homografi. Vektor kolom \mathbf{tvec} didapatkan dengan menyelesaikan persamaan linier $\mathbf{X} \mathbf{tvec} = \mathbf{U}$. Apabila matriks \mathbf{X} adalah matriks bujur sangkar dan determinannya sama dengan nol, maka kalibrasi kamera tidak bisa dilakukan dan harus

diganti data lain. Apabila matriks \mathbf{X} adalah matriks bujur sangkar dan determinannya tidak sama dengan nol maka $\mathbf{tvec} = \mathbf{X}^{-1}\mathbf{U}$. Apabila \mathbf{X} bukan merupakan matriks bujur sangkar, maka sebelum \mathbf{X} diinverskan harus dikali dengan \mathbf{X}^T agar menjadi matriks bujur sangkar, kemudian hasil perkalian \mathbf{X}^T dengan \mathbf{X} diinverskan. Matriks \mathbf{U} juga harus dikali dengan \mathbf{X}^T , karena \mathbf{X} dikali \mathbf{X}^T . Apabila hasil perkalian \mathbf{X}^T dengan \mathbf{X} determinannya sama dengan nol, maka kalibrasi kamera tidak bisa dilakukan, sehingga data yang digunakan harus diganti dengan data lain. Apabila perkalian \mathbf{X}^T dengan \mathbf{X} determinannya tidak sama dengan nol, maka untuk mendapatkan vektor kolom \mathbf{tvec} menggunakan persamaan (4.3) berikut ini,

$$\mathbf{tvec} = (\mathbf{X}^T \mathbf{X})^{-1}(\mathbf{X}^T \mathbf{U}) = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{bmatrix} \quad (4.3)$$

Apabila matriks \mathbf{tvec} sudah didapatkan, elemen matriks \mathbf{tvec} ditambah satu elemen lagi yaitu $P_9 = 1$, kemudian dapat disusun matriks \mathbf{T} seperti pada persamaan (2.3). Setiap elemen matriks \mathbf{T} dibagi dengan $\mathbf{T}(3,3)$. Matriks \mathbf{T} yang setiap elemennya telah dibagi dengan $\mathbf{T}(3,3)$ harus ditranpose untuk membentuk matriks homografi seperti pada persamaan (2.4).

4. Menghitung matriks \mathbf{V} dan \mathbf{B}

Matriks homografi yang telah didapatkan, digunakan untuk menghitung matriks \mathbf{V} menggunakan persamaan (4.4) berikut ini,

$$\mathbf{V} = \begin{bmatrix} h_{i,1}h_{j,1} \\ h_{i,1}h_{j,2} + h_{i,2}h_{j,1} \\ h_{i,2}h_{j,2} \\ h_{i,3}h_{j,1} + h_{i,1}h_{j,3} \\ h_{i,3}h_{j,2} + h_{i,2}h_{j,3} \\ h_{i,3}h_{j,3} \end{bmatrix}^T \quad (4.4)$$

Matriks \mathbf{V} didekomposisi dengan metode *Singular Value Decomposition* (SVD) dan dimisalkan \mathbf{D} adalah singular vektor kanan dari \mathbf{V} . Bentuk dari matriks \mathbf{D} telah ditunjukkan pada persamaan (2.5). Matriks kolom terakhir dari matriks \mathbf{D} , merupakan penyusun dari matriks \mathbf{B} . Bentuk matriks \mathbf{B} ditunjukkan pada persamaan (2.7).

5. Menghitung parameter intrinsik

Matriks \mathbf{B} yang didapatkan digunakan untuk menghitung parameter-parameter intrinsik. Perhitungan parameter-parameter intrinsik menggunakan persamaan (2.8-2.13). Parameter-parameter intrinsik disusun menjadi matriks \mathbf{A} seperti pada persamaan (2.14).

6. Menghitung Parameter Ekstrinsik

Langkah pertama dalam menghitung parameter-parameter ekstrinsik yang berupa parameter translasi dan rotasi adalah dengan memisalkan matriks kolom pertama matriks homografi dengan \mathbf{h}_1 , matriks kolom sebagai kedua \mathbf{h}_2 dan ketiga \mathbf{h}_3 seperti pada persamaan (4.5). Sehingga dapat dihitung parameter-parameter ekstrinsik dengan persamaan (2.15-2.22),

$$\mathbf{H} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (4.5)$$

Parameter ekstrinsik yang dihasilkan, selanjut akan dioptimalkan, sehingga untuk menghemat komputasi matriks \mathbf{R} dikonversi menjadi vektor 3×1 . Konversi matriks rotasi menggunakan persamaan (4.6) berikut ini,

$$\mathbf{R2} = \frac{1}{2 \sin(\zeta)} \begin{bmatrix} \mathbf{R}(3,2) - \mathbf{R}(2,3) \\ \mathbf{R}(1,3) - \mathbf{R}(3,1) \\ \mathbf{R}(2,1) - \mathbf{R}(1,2) \end{bmatrix} \zeta \quad (4.6)$$

$$\zeta = \arccos\left(\frac{\text{trace}(\mathbf{R1})-1}{2}\right).$$

7. Mengoptimalkan Semua Parameter agar Errornya Kecil dan Mengkonversi kembali Vektor Rotasi menjadi Matriks Rotasi

Setiap parameter didapatkan harus dioptimalkan. Cara mengoptimalkan parameter yang ada adalah dengan menggunakan algoritma Levenberg-Marquadt, dimana dimisalkan ada n citra dan ada m titik pada setiap citra, dengan persamaan (4.7) berikut parameter-parameter hasil kalibrasi kamera zhang dioptimalkan,

$$\sum_{i=1}^n \sum_{j=1}^m \| \mathbf{m}_{ij} - \check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j) \|^2 \quad (4.7)$$

dimana $\check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)$ adalah proyeksi dari titik \mathbf{M}_j di citra- i , menurut persamaan $s\mathbf{m} = \mathbf{H}\mathbf{M}$ dengan s adalah sembarang faktor skala

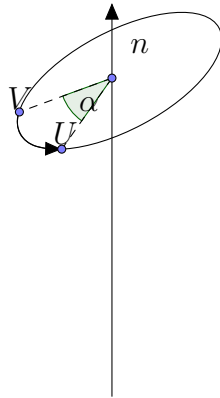
dan $\mathbf{H} = \mathbf{A}[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]$. k_1 dan k_2 adalah koefisien distorsi radial, sebagai inisialisasi sama dengan nol. Matriks rotasi yang telah dioptimalkan juga akan dikembalikan ke bentuk semula, yaitu matriks 3×3 . Konversi vektor 1×3 menjadi matriks 3×3 menggunakan persamaan rotasi rodriques yang ditampilkan pada persamaan (4.8) berikut ini,

$$\mathbf{R}_k = \mathbf{I} \cos(\alpha) + ([\mathbf{n}]_{\times}^2 + \mathbf{I}) (1 - \cos(\alpha)) + [\mathbf{n}]_{\times} \sin(\alpha) \quad (4.8)$$

$k = 1, \dots, n$. n adalah banyaknya matriks baris dan α adalah norm dari $\mathbf{R2}$, sedangkan $[\mathbf{n}]_{\times}$ adalah matriks sebagai berikut,

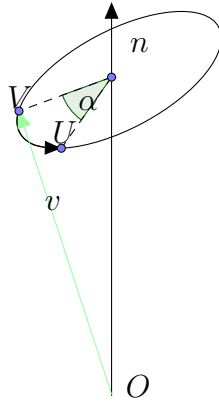
$$[\mathbf{n}]_{\times} = \begin{bmatrix} 0 & -\mathbf{R2}(3) & \mathbf{R2}(2) \\ \mathbf{R2}(3) & 0 & -\mathbf{R2}(1) \\ -\mathbf{R2}(2) & \mathbf{R2}(1) & 0 \end{bmatrix}$$

Persamaan rotasi Rodriguez didapatkan dengan memisalkan titik V yang akan dirotasi menuju U dengan sudut α di sekitar sumbu \mathbf{n} , seperti pada Gambar 4.6 berikut ini (Jiri, 13 Agustus 2012),



Gambar 4.6: Titik V dirotasi ke titik U dengan sudut α

Langkah selanjutnya, didefinisikan sebuah vektor \mathbf{v} dari titik asal O ke titik V dan titik V tegak lurus terhadap sumbu rotasi \mathbf{n} . Vektor \mathbf{n} juga dinormalkan untuk mewakili sumbu rotasi $|\mathbf{n}| = 1$. Simulasinya seperti pada Gambar 4.7 berikut ini,



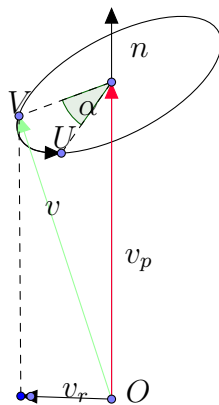
Gambar 4.7: Vektor \mathbf{v} dari titik asal ke titik V

Dimisalkan vektor \mathbf{v}_p yang merupakan proyeksi pada sumbu rotasi \mathbf{n} , simulasinya seperti pada Gambar 4.8. Nilai \mathbf{v}_p dihitung sebagai proyeksi vektor, sehingga dengan menggunakan proyeksi vektor \mathbf{v} dikalikan titik dengan vektor satuan \mathbf{n} arah \mathbf{n} .

$$\mathbf{v}_p = (\mathbf{n} \cdot \mathbf{v})\mathbf{n}$$

\mathbf{v}_r tegak lurus dengan proyeksi \mathbf{v}_p , sehingga dengan rumus pythagoras didapatkan vektor \mathbf{v}_r adalah,

$$\mathbf{v}_r = \mathbf{v} - \mathbf{v}_p$$

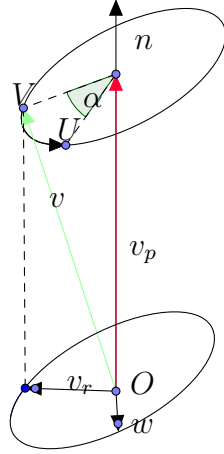


Gambar 4.8: Memisalkan vektor \mathbf{v}_p dan \mathbf{v}_r

Sekarang ingin dicari vektor yang tegak lurus dengan \mathbf{v} dan \mathbf{v}_p , sehingga

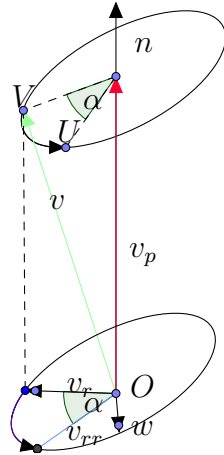
dimisalkan vektor \mathbf{w} tegak lurus terhadap \mathbf{v} dan \mathbf{v}_p seperti pada Gambar 4.9. Oleh karena itu, nilai vektor \mathbf{w} menjadi,

$$\mathbf{w} = \mathbf{n} \times \mathbf{v}$$



Gambar 4.9: Memisalkan vektor \mathbf{w}

Sebelum melakukan langkah selanjutnya, dicoba untuk merotasi \mathbf{v}_r dengan sudut 90° menuju \mathbf{w} . Misalkan $\mathbf{v}_r = [v_{rx}, v_{ry}]^T$ dan $\mathbf{w}_r = [w_x, w_y]^T$, dengan rumus rotasi didapatkan $\mathbf{w}_x = -\mathbf{v}_{ry}$ dan $\mathbf{w}_y = \mathbf{v}_{rx}$. Selanjutnya dilakukan rotasi lagi dari \mathbf{v}_r dengan sudut α dimisalkan vektor baru \mathbf{v}_{rr} seperti pada Gambar 4.10 berikut ini,

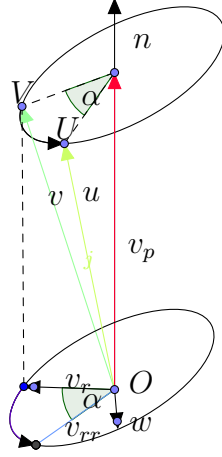


Gambar 4.10: Memisalkan vektor \mathbf{v}_{rr}

Menggunakan rumus rotasi didapatkan \mathbf{v}_{rr} sebagai berikut,

$$\mathbf{v}_{rr} = \mathbf{v}_r \cos \alpha + \mathbf{w} \sin \alpha$$

Ada satu langkah lagi untuk mendapatkan titik O ke titik U , yaitu dengan menambahkan vektor \mathbf{u} diantara \mathbf{v}_{rr} dan \mathbf{v}_p seperti pada Gambar 4.11 berikut ini,



Gambar 4.11: Memisalkan vektor \mathbf{u}

Karena \mathbf{v}_{rr} tegak lurus dengan \mathbf{v}_p , sehingga persamaan untuk mendapatkan \mathbf{u} menjadi,

$$\mathbf{u} = \mathbf{v}_{rr} + \mathbf{v}_p \quad (4.9)$$

$$= \mathbf{v} \cos \alpha + (\mathbf{n} \cdot \mathbf{v})\mathbf{n}(1 - \cos \alpha) + (\mathbf{n} \times \mathbf{v}) \sin \alpha \quad (4.10)$$

Persamaan (4.9) dapat disederhanakan dengan mengubah perkalian titik dan silang, dengan asumsi bahwa semua vektor adalah vektor kolom, sehingga perkalian titik dihitung sebagai,

$$\mathbf{v}_p = (\mathbf{n} \cdot \mathbf{v})\mathbf{n} = \mathbf{n}\mathbf{n}^T \mathbf{v}$$

Penyederhanaan perkalian silang dilakukan dengan menggunakan trik matriks skew-simetris. Pertama-tama vektor \mathbf{n} dirubah menjadi matriks skew-simetris seperti pada persamaan (4.11) dan kemudian mengalikan dengan vektor \mathbf{v} .

$$[\mathbf{n}]_{\times} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \quad (4.11)$$

$$\mathbf{w} = \mathbf{n} \times \mathbf{v} = [\mathbf{n}]_{\times} \mathbf{v}$$

Mensubstitusikan perubahan perkalian titik dan silang persamaan, sehingga didapatkan rumus Rodriguez.

$$\mathbf{u} = \mathbf{v} \cos \alpha + \mathbf{nn}^T \mathbf{v} (1 - \cos \alpha) + [\mathbf{n}]_{\times} \mathbf{v} \sin \alpha$$

Matriks rotasi \mathbf{R} didapatkan dengan membagi persamaan \mathbf{u} menjadi komponen \mathbf{R} dan \mathbf{v} seperti persamaan berikut ini,

$$\mathbf{u} = \mathbf{R}\mathbf{v}$$

Sehingga didapatkan matriks \mathbf{R} ,

$$\begin{aligned} \mathbf{u} &= (\mathbf{I} \cos \alpha + \mathbf{nn}^T (1 - \cos \alpha) + [\mathbf{n}]_{\times} \sin \alpha) \mathbf{v} \\ \mathbf{R} &= \mathbf{I} \cos \alpha + \mathbf{nn}^T (1 - \cos \alpha) + [\mathbf{n}]_{\times} \sin \alpha \end{aligned}$$

Matriks \mathbf{R} yang dihasilkan masih belum sama dengan persamaan (4.8), oleh sebab itu dengan menggunakan identitas perkalian *outer*,

$$\mathbf{nn}^T = [\mathbf{n}]_{\times}^2 + \mathbf{I}$$

\mathbf{I} adalah matriks identitas 3×3 . Hal ini dapat dibuktikan dengan mengikuti beberapa langkah. Pertama-tama dituliskan \mathbf{nn}^T sebagai berikut,

$$\begin{aligned} \mathbf{nn}^T &= \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \\ &= \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} \end{aligned}$$

Jika matriks $[\mathbf{n}]_{\times}$ dan dikuadratkan akan menjadi seperti berikut ini,

$$[\mathbf{n}]_{\times}^2 = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

$$[\mathbf{n}]_{\times}^2 = \begin{bmatrix} -n_3^2 - n_2^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & -n_3^2 - n_1^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & -n_2^2 - n_1^2 \end{bmatrix}$$

Sekarang dapat dilihat bahwa itu benar-benar terlihat seperti \mathbf{nn}^T . Langkah terakhir adalah mengubah diagonal dari $[\mathbf{n}]_{\times}^2$ menjadi diagonal \mathbf{nn}^T . Misalkan,

$$n_1^2 + n_2^2 + n_3^2 = 1 \Rightarrow n_1^2 = -n_2^2 - n_3^2 + 1$$

Oleh karena itu dapat ditulis

$$\begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} = \begin{bmatrix} -n_3^2 - n_2^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & -n_3^2 - n_1^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & -n_2^2 - n_1^2 \end{bmatrix} + \mathbf{I}$$

Oleh karena itu didapatkan,

$$\mathbf{nn}^T = [\mathbf{n}]_{\times}^2 + \mathbf{I}$$

Persamaan rotasi Rodriguez menjadi,

$$\mathbf{R}_k = \mathbf{I} \cos(\alpha) + ([\mathbf{n}]_{\times}^2 + \mathbf{I}) (1 - \cos(\alpha)) + [\mathbf{n}]_{\times} \sin(\alpha) \quad (4.12)$$

$k = 1, \dots, n$, n adalah banyaknya matriks baris.

8. Menghilangkan Distorsi Citra

Distorsi adalah penyimpangan dari proyeksi bujursangkar. Distorsi pada citra dapat mempengaruhi estimasi jarak yang dihasilkan, oleh karena itu digunakan distorsi radial untuk memperbaiki penyimpangan pada proyeksi bujur sangkar. Distorsi radial cukup simetris dalam memperbaiki penyimpangan.

Langkah pertama dalam menghilangkan distorsi citra adalah dengan mengetahui ukuran citra. Jika citra memiliki lebar misalkan i dan tinggi misalkan j , maka dibutuhkan misal \mathbf{q} matriks kolom berukuran $((j \times i) \times 1)$ dengan semua elemen 1. Langkah selanjutnya didefinisikan \mathbf{p} , yaitu matriks kolom dengan elemen urutan 1 sampai i sebanyak j , misal sebuah matriks memiliki lebar 2 dan tinggi 3, sehingga matriks kolom \mathbf{p} adalah

$$\mathbf{p} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

Misalkan matriks \mathbf{X} sebagai berikut,

$$\mathbf{X} = \mathbf{A}^{-1}[\mathbf{q} \ \mathbf{p} \ \mathbf{q}]$$

Persamaan untuk menghilangkan distorsi citra dengan koefisien distorsi radial adalah sebagai berikut,

$$\mathbf{x} = [\mathbf{X}(1, :)] \times (1 + k_1 \mathbf{r} + k_2 [\mathbf{r}]^2) \quad (4.13)$$

$$\mathbf{y} = [\mathbf{X}(2, :)] \times (1 + k_1 \mathbf{r} + k_2 [\mathbf{r}]^2) \quad (4.14)$$

$\mathbf{r} = [\mathbf{X}(1, :)]^2 + [\mathbf{X}(2, :)]^2$. Setelah didapatkan \mathbf{x} dan \mathbf{y} , selanjutnya dihitung $\mathbf{u} = \gamma \times \mathbf{x} + u_0$ dan $\mathbf{v} = \beta \times \mathbf{y} + v_0$. Langkah terakhir adalah dengan menginterpolasi \mathbf{u} dan \mathbf{v} dengan interpolasi *nearest neighbor*, sehingga didapatkan citra tak terdistorsi.

4.3.2 Perancangan GMM untuk Estimasi Jarak pada Obyek Bergerak

Tahap ini dilakukan perancangan GMM untuk estimasi jarak pada obyek yang bergerak. Sebelum masuk dalam perancangan GMM, terlebih dahulu dilakukan morfologi citra untuk mempermudah dalam pemisahan *background* dan *foreground*. Ada enam macam morfologi citra yang bisa digunakan. Morfologi citra yang pertama adalah filtering untuk mereduksi noise. Kemudian yang kedua adalah erosi, untuk memperkecil obyek agar noise-noise kecil yang ada pada citra hilang. Kemudian yang ketiga adalah dilasi, untuk memperbesar kembali obyek yang tersisa. Terakhir dilakukan filling untuk mengisi bagian dalam dari obyek dengan angka satu, sedangkan di luar obyek diisi satu. Apabila morfologi citra telah dilakukan, maka dapat dilanjutkan ke dalam langkah-langkah perancangan GMM berikut ini:

1. Tahap inisialisasi dan Seleksi Input

Langkah pertama dalam perancangan GMM adalah memberikan nilai awal varian (σ^2), banyaknya pelatihan frame, laju pelatihan (α),

banyaknya model *background*. Pertama-tama menghitung probabilitas piksel \mathbf{C}^n vektor dari citra RGB pada waktu ke- n menggunakan persamaan (4.15) berikut ini (Stauffer, Grimson, 1999),

$$p(\mathbf{C}^n) = \sum_{j=1}^k \omega_k \eta(\mathbf{C}^n | \theta_j) \quad (4.15)$$

dimana ω_k adalah bobot dari komponen gaussian ke- k . η adalah distribusi normal komponen ke- k yang dihitung menggunakan persamaan berikut ini,

$$\eta(\mathbf{C} | \theta_j) = \eta(\mathbf{C} | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{C} - \mu_k)^T \Sigma_k^{-1} (\mathbf{C} - \mu_k)}$$

μ_k adalah mean dan $\Sigma_k = \sigma^2 \mathbf{I}$ yang merupakan kovarian ke- k . $|\Sigma|$ adalah determinan dari kovarian. Distribusi pada persamaan (4.16) adalah distribusi untuk menentukan update parameter, apabila tidak terpenuhi maka M^{n+1} bernilai nol dan parameter yang diperbarui hanya parameter bobot. Apabila terpenuhi maka M^{n+1} bernilai satu dan parameter yang diperbarui adalah bobot, mean dan kovarian.

$$\mu_t - 2.5 * \sigma_k < \mathbf{C}^n < \mu_t + 2.5 * \sigma_k \quad (4.16)$$

2. Tahap Memperbarui Parameter

Tahap pertama telah menghasilkan parameter apa saja yang diperbarui. Persamaan yang digunakan untuk memperbarui nilai dari ω_k^{n+1} (bobot), μ_k^{n+1} (mean) dan kovarian Σ_k ditampilkan pada persamaan-persamaan berikut ini (Kaewtrakulpong, Bowden, 2001),

$$\omega_k^{n+1} = (1 - \alpha) \omega_k^n + \alpha(M^{n+1}) \quad (4.17)$$

$$\Sigma_k^{n+1} = (1 - \alpha) \Sigma_k^n + \rho(\mathbf{C}^{n+1} - \mu_k^{n+1})(\mathbf{C}^{n+1} - \mu_k^{n+1})^T \quad (4.18)$$

$$\mu_k^{n+1} = (1 - \alpha) \mu_k^n + \rho \mathbf{C}^{n+1}$$

$$\rho = \alpha \eta(\mathbf{C}^{n+1} | \mu_k^n, \Sigma_k^n)$$

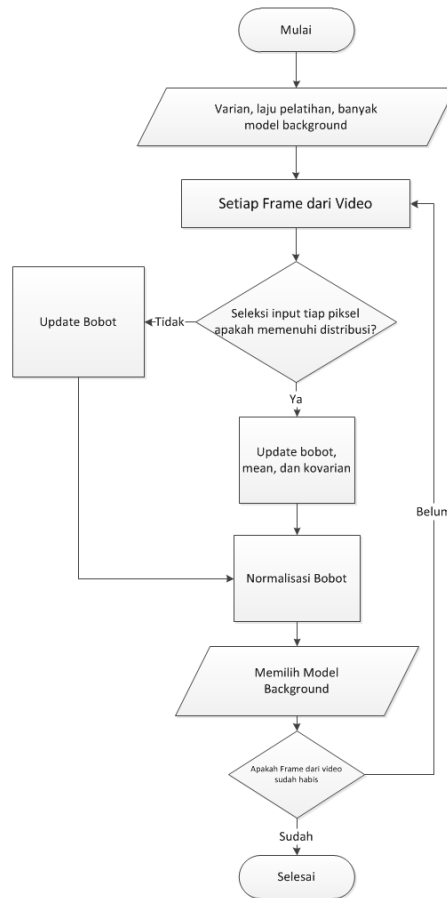
3. Tahap pemilihan distribusi yang mencerminkan *background*

Langkah selanjutnya adalah memilih model yang menyerupai *background*. Setiap piksel dalam *scene* dimodelkan dengan K distribusi gaussian. Model akan menormalisasi bobot atau dengan kata lain mengurutkan model berdasarkan $\frac{\omega}{\sigma^2}$, dengan distribusi yang paling menyerupai *background* (B) berada di bagian atas. Persamaan untuk memilih B distribusi pertama yang dijadikan *background* adalah,

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k < L \right) \quad (4.19)$$

Treshold L adalah nilai minimum yang harus dipenuhi oleh model *background*.

Berikut alur deteksi obyek bergerak pada Gambar 4.12,



Gambar 4.12: Flowcart deteksi obyek bergerak

Obyek yang bergerak dideteksi terlebih dahulu, selanjutnya jarak dari obyek ke kamera diestimasi. Salah satu informasi yang perlu diketahui dalam estimasi jarak adalah jarak antara dua kamera yang atau *baseline* (b). Koordinat ujung paling kiri obyek yang terdeteksi pada kamera kiri maupun kanan dimisalkan (x_l, y_l) dan (x_r, y_r) . Langkah selanjutnya estimasi jarak dari kamera ke obyek menggunakan persamaan (4.20),

$$\text{Jarak} = \frac{b \times f}{x_l - x_r} \quad (4.20)$$

f panjang fokus, sedangkan $(x_l - x_r)$ adalah dispariti. Panjang fokus dihitung menggunakan persamaan berikut

$$\text{Panjang fokus} = \frac{\text{jarak} \times \text{dispariti}}{\text{baseline}}$$

Waktu komputasi dapat dihemat dengan menyederhanakan perhitungan faktor, yang didefinisikan pada persamaan (4.21) (Hantao, 2005).

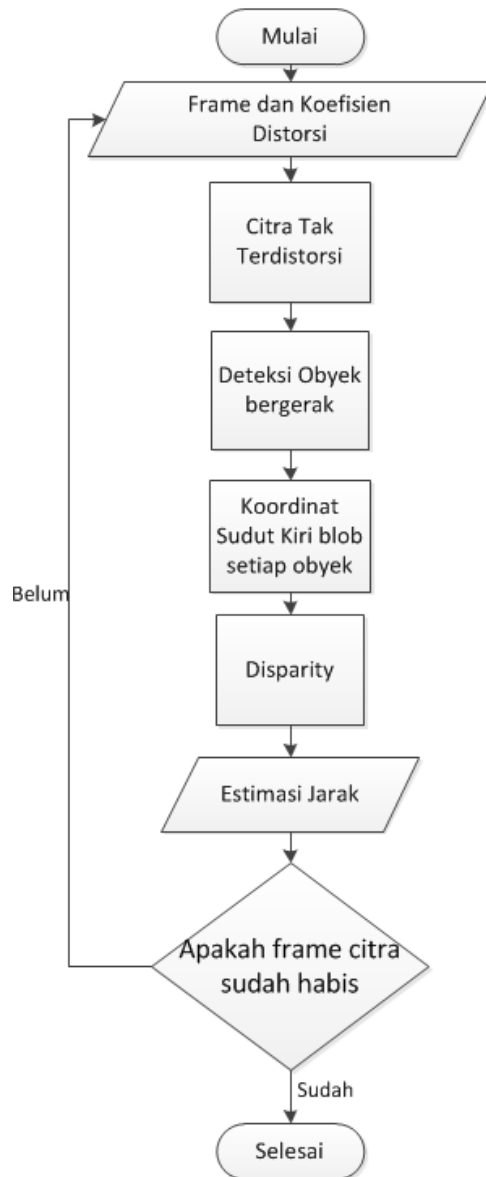
$$\text{Faktor} = \text{baseline} \times \text{panjang fokus} = \text{Jarak asli} \times \text{dispariti} \quad (4.21)$$

Persamaan Estimasi jarak menjadi seperti (4.22) berikut ini,

$$\text{Estimasi jarak} = \frac{\text{rata} - \text{rata faktor}}{\text{dispariti}} \quad (4.22)$$

4.3.3 Perancangan Uji Coba Parameter-Parameter hasil Kalibrasi Kamera pada Deteksi Obyek Bergerak

Parameter ekstrinsik dan intrinsik yang didapatkan pada kalibrasi kamera merupakan representasi dari posisi kamera dan pola planar ketika pengambilan data. Parameter ekstrinsik dan intrinsik yang dihasilkan harus di uji coba untuk mengetahui mengevaluasi proses kalibrasi yang telah dilakukan, juga untuk mengetahui kualitas parameter-parameter yang dihasilkan. Uji coba dilakukan dengan mengestimasi koefisien distorsi menggunakan parameter ekstrinsik dan intrinsik yang telah dihasilkan sebelumnya. Koefisien distorsi yang telah dihasilkan digunakan untuk menghilangkan distorsi pada setiap frame, kemudian diestimasi jarak obyek yang bergerak. Gambar 4.13 berikut ini flowcart estimasi jarak,



Gambar 4.13: Flowcart estimasi jarak

4.4 Implementasi Menggunakan Perangkat Lunak

Tahap selanjutnya yang perlu dilakukan setelah merancang langkah-langkah perangkat lunak adalah mengimplementasikan langkah-langkah tersebut kedalam program yang dikembangkan menggunakan MATLAB. Sub bab ini akan dijelaskan mengenai implementasi perangkat lunak yang akan dibangun.

4.4.1 Lingkungan Implementasi

Lingkungan implementasi perangkat lunak yang akan dibangun pada tesis ini meliputi perangkat keras dan perangkat lunak yang digunakan. Detailnya

diberikan sebagai berikut:

Perangkat Keras:

- Prosesor PC : Intel(R) Core i5 2400S dengan Memory PC RAM 4.00 G
- Kamera : Logitech merek C170 5 MP, resolusi XVGA (1024 x 768)

Perangkat Lunak:

- Sistem Operasi : Microsoft Windows 7 Ultimate 64-bit
- Tools Pembangun : MATLAB

4.4.2 Perancangan Kalibrasi Kamera Zhang

Langkah lanjutan yang akan dilakukan setelah proses analisis sistem dan perancangan perangkat lunak adalah perancangan kalibrasi kamera zhang yang akan dibangun pada Tesis ini. Perancangan ini terdiri atas beberapa proses yang akan dijelaskan dibawah ini.

1. Memasukkan pola planar

Citra pola planar adalah komponen terpenting dalam melakukan ekstraksi parameter-parameter intrinsik, ekstrinsik menggunakan algoritma kalibrasi kamera Zhang. Gambar pola planar yang sudah dicetak dipapan baik dari kamera kiri maupun kamera kanan ditunjukkan pada Gambar 4.3 dan Gambar 4.4.

2. Deteksi titik pada setiap pola planar

Posisi pola planar yang digunakan, memberi orientasi tersendiri pada proses estimasi parameter ekstrinsik dan intrinsik, semakin banyak kombinasi posisi yang dipakai, akan memberikan orientasi yang banyak pula. Setiap titik pada pola planar harus dideteksi untuk mendapatkan koordinat titik yang akan digunakan pada perhitungan titik dunia dan matriks homografi.

3. Menghitung Homografi

Matriks homografi merupakan korelasi diantara dua citra, untuk menghitung matriks homografi dibutuhkan matriks (x, y) dan (u, v) yaitu matriks titik dunia dan titik pola planar dalam citra, proses menghitung homografi dilakukan dengan menggunakan algoritma 1 berikut ini,

Algoritma 1 Menghitung Homografi

Input : (x, y) dan (u, v)
Menghitung \mathbf{X}
 $\mathbf{U} = [u^T; v^T]$
Menyelesaikan $\mathbf{X} \quad \mathbf{tvec} = \mathbf{U}$
 $\mathbf{tvec}(9) = 1$
Menghitung matriks \mathbf{T} seperti pada persamaan (2.3)
Pada matriks \mathbf{T} setiap elemennya dikali $\mathbf{T}(3, 3)$
Tranpose matriks \mathbf{T} yang setiap elemennya sudah dikali $\mathbf{T}(3, 3)$

4. Menghitung matriks \mathbf{V} dan \mathbf{B}

Matriks homografi yang didapatkan digunakan untuk menghitung matriks \mathbf{V} dan \mathbf{B} . Matriks \mathbf{V} disusun dari matriks homografi sedangkan matriks \mathbf{B} disusun dari hasil mendekomposisi matriks \mathbf{V} yaitu dengan menggunakan matriks kolom terakhir dari singular vektor kanan dari \mathbf{V} . Proses menghitung matriks \mathbf{V} dan \mathbf{B} dilakukan dengan menggunakan algoritma 2 berikut ini,

Algoritma 2 Menghitung matriks \mathbf{V} dan \mathbf{B}

$(\sim, \sim, \mathbf{D}) = \text{svd}(\mathbf{V})$
 $[x, y] = \text{size}(\mathbf{D})$
 $k = 1$
for $i = 1 : y$ **do**
 for $j = 1 : x$ **do**
 $\mathbf{B}(i, j) = \mathbf{U}(k, y)$
 $k = k + 1$
 end for
end for

5. Menghitung parameter intrinsik

Pada fase ini parameter intrinsik yang meliputi $v_0, \lambda, \theta, \gamma$ dan u_0 dihitung menggunakan matriks \mathbf{B} yang telah dihasilkan pada langkah sebelumnya. Parameter intrinsik dihitung menggunakan persamaan (2.8, 2.9, 2.10, 2.11, 2.12, 2.13), kemudian menyusun parameter-parameter intrinsik menjadi matriks \mathbf{A} seperti pada persamaan (2.14). Parameter-parameter intrinsik disusun menjadi matriks \mathbf{A} , karena dalam perhitungan ekstrinsik membutuhkan parameter-parameter intrinsik dalam bentuk ini.

6. Menghitung parameter ekstrinsik

Parameter ekstrinsik yang meliputi translasi dan rotasi dihitung dengan menggunakan matriks homografi dan \mathbf{A} . Matriks rotasi harus diubah menjadi vektor rotasi 1×3 menggunakan persamaan rotasi Rodriguez.

Proses menghitung parameter ekstrinsik dilakukan dengan menggunakan algoritma 3 berikut ini,

Algoritma 3 Menghitung Parameter Ekstrinsik

```

Input : Matriks Homografi, Matriks A
 $n = \text{size}((\text{homografi}), 3)$ 
 $A_{inv} = \text{inv}(A)$ 
for  $i = 1 : n$  do
    for  $k = 1 : 3$  do
         $h_1(1, k) = H(k, 1)$ 
         $h_2(1, k) = H(k, 2)$ 
         $h_3(1, k) = H(k, 3)$ 
         $\alpha = 1/(\|A_{inv}\| h_1)$ 
         $\mathbf{r}_1 = \alpha A_{inv} h_1$ 
         $\mathbf{r}_2 = \alpha A_{inv} h_2$ 
         $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$ 
         $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$ 
        Konversi matriks R berukuran  $3 \times 3$  menjadi vektor rotasi  $1 \times 3$ 
    end for
end for
 $\mathbf{t} = \text{lambda } A_{inv} h_3$ 

```

7. Mengoptimalkan Semua Parameter dan Mengkonversi Vektor Rotasi menjadi Matriks Rotasi

Parameter ekstrinsik dan intrinsik harus dioptimalkan dalam setiap iterasi untuk memperkecil error. Pertama-tama setiap kamera dioptimalkan masing-masing setelah itu semua parameter dioptimalkan bersama-sama dengan persamaan (4.7). Parameter-parameter rotasi yang telah dioptimalkan, harus dikembalikan lagi menjadi matriks berukuran 3×3 lagi menggunakan persamaan (4.8).

4.4.3 Perancangan Estimasi Jarak pada Obyek Bergerak

1. Memanggil file video dan membaca setiap frame dalam sekuen video
Perangkat lunak akan memanggil file video yang diinginkan dan memecah setiap frame dalam video.
2. Menghilangkan distorsi citra
Setiap frame yang akan diproses ke dalam algoritma GMM, maka harus dihilangkan terlebih dahulu distorsinya agar tidak terjadi kesalahan saat menghitung dispariti.
3. Morfologi Citra
Morfologi citra dilakukan untuk mempermudah dalam pemisahan *foreground* dan *background*.

4. Menyusun GMM

Proses menyusun algoritma GMM dilakukan dengan menggunakan algoritma 4 berikut ini,

Algoritma 4 Algoritma GMM

Memberikan nilai awal varian, laju pelatihan, maksimum iterasi pelatihan, GPD dan maksimum *background*

for $i = 1 : \text{maksimum iterasi pelatihan}$ **do**

 Menghitung σ_t^2

 Menghitung $\Sigma_{k,t}$

 Menghitung $\eta(\mathbf{C}, \mu, \Sigma)$

 Menghitung μ_t

 Menghitung ρ

 Menghitung $\omega_{k,t}$

end for

for $i = 1 : \text{frame terakhir}$ **do**

if $\mathbf{C} > \mu_k - 2.5 * \sigma$ dan $\mathbf{C} < \mu_k + 2.5 * \sigma$ **then**

 Memilih *background* $B_{\text{background}}$

else

Return

end if

end for

5. Menandai setiap obyek yang terdeteksi.

Setiap obyek yang sudah terdeteksi sebagai background akan menjadi warna hitam, sedangkan yang berwarna putih adalah sisanya sebagai *foreground*. Citra yang masih berwarna putih diberi kotak agar membedakannya sebagai *foreground*.

6. Mengestimasi jarak obyek yang terdeteksi.

Obyek yang terdeteksi diestimasi jaraknya dengan persamaan (4.22).

BAB 5

UJI COBA DAN PEMBAHASAN

Bab ini akan melakukan uji coba program kalibrasi kamera zhang dan implementasinya pada penghilangan distorsi citra dan deteksi obyek bergereak. Pengujian ini dilakukan untuk menguji parameter-parameter yang dihasilkan dari kalibrasi kamera.

5.1 Ruang Lingkup UJI COBA

Ruang lingkup uji coba dan implementasi program yang dibangun dalam tesis ini meliputi perangkat keras dan perangkat lunak yang digunakan. Perangkat kerasnya terdiri PC intel(R) core i5 2400S dengan memory 4.00 GB dan kamera web cam merek logitech tipe C170 5 mega piksel dengan resolusi XVGA (1024 x 768). Kamera logitech C170 dipilih karena kamera ini mampu memenuhi kebutuhan dalam tesis ini, selain itu kamera ini merupakan salah satu jenis kamera yang bisa dikoneksikan dengan tools pembangun yang digunakan, yaitu Matlab. Perangkat lunak merupakan sistem operasi dari PC, yaitu microsoft windows 7 ultimate 64-bit. Perangkat lunak lainnya berupa tools pembangun, yaitu matlab.

Berdasarkan ruang lingkup yang telah dipaparkan, perangkat pembangun yang dibangun pada tesis ini tidak dipaket (*package*) menjadi sebuah perangkat lunak *portable* yang dapat dijalankan di sebarang sistem operasi, maka uji coba program hanya dapat dijalankan pada sistem operasi yang telah terinstal matlab.

5.2 Persiapan sebelum Uji Coba

Beberapa hal perlu difikirkan dalam melakukan uji coba, terutama mengenai lokasi pengambilan data juga mengenai baju yang dipakai oleh model dalam deteksi jarak. Lokasi yang digunakan adalah yang tidak menyusahkan dalam pengambilan data maupun pengolahan data. Penelitian ini dibatasi pada obyek tunggal yang diestimasi jaraknya, maka dibutuhkan lokasi yang sepi, sehingga gangguan berupa obyek lain yang masuk bisa dicegah sejak awal. Faktor selanjutnya yang perlu difikirkan adalah mengenai baju yang digunakan oleh model, karena dalam penelitian ini pemisahan *background* menggunakan warna hitam dan *foreground* menggunakan warna putih, maka disarankan pada model untuk tidak menggunakan warna hitam agar tidak terdeteksi sebagai *background*. Faktor selanjutnya yang perlu difikirkan adalah ketika pengambilan data di lapangan, beberapa peralatan juga perlu dibawa, diantaranya alat pengukur jarak, pola planar, kamera stereo dan laptop. Alat pengukur jarak digunakan untuk mengukur jarak sesungguhnya, sehingga dapat dihitung rasio error antara jarak sesungguhnya

dengan jarak estimasi. Pola planar adalah pola yang digunakan untuk melakukan kalibrasi kamera zhang. Barang terpenting yang perlu dibawa adalah laptop digunakan untuk menyimpan data dan kamera stereo adalah alat yang digunakan mengambil data. Gambar 5.1 berikut ini adalah kamera stereo yang digunakan pada penelitian ini,



Gambar 5.1: Dua web cam yang disusun menjadi satu sistem kamera stereo

Pola planar inilah yang digunakan dalam kalibrasi kamera untuk mendapatkan parameter ekstrinsik dan intrinsik kamera. Berikut ini Gambar 5.2, pola planar yang digunakan,



Gambar 5.2: Papan pola planar yang digunakan

Terakhir kamera stereo yang telah disusun, ditempatkan pada tempat yang sesuai untuk merekam obyek yang diinginkan.

5.3 UJI COBA

Sub-bab ini menyajikan perhitungan kalibrasi kamera zhang menggunakan Gambar 4.3 dan Gambar 4.4. Langkah pertama, citra yang dihasilkan pada Gambar 4.3 dan Gambar 4.4 dideteksi setiap titik pada pola planar, hasilnya

dapat dilihat pada Gambar 5.3 dan Gambar 5.4.



Gambar 5.3: Deteksi titik pada pola planar, pada citra dari kamera kiri



Gambar 5.4: Deteksi titik pada pola planar, pada citra dari kamera kanan

Koordinat setiap titik pada pola planar baik dari kamera kiri maupun kanan dimisalkan **titikCitra**, koordinat dalam **titikCitra** ditampilkan beberapa pada Tabel 5.1 berikut ini, data selengkapnya begitu juga perhitungannya ditampilkan pada lampiran A,

Tabel 5.1: Koordinat Titik Pola Planar Kiri dan Kanan

No Titik	Koordinat Titik Pola Kiri	Koordinat Titik Pola Kanan
Titik 1	(198.2897, 202.9816)	(142.3913, 276.2516)
Titik 2	(196.9440, 241.0734)	(140.1024, 313.1302)
Titik 3	(195.3491, 279.7053)	(137.7794, 350.3179)
Titik 4	(193.9997, 318.3077)	(135.2941, 388.1032)
⋮	⋮	⋮
Titik 51	(505.8939, 288.0714)	(437.6999, 359.9727)
Titik 52	(505.2487, 327.3213)	(437.0393, 398.0932)
Titik 53	(504.7968, 366.8463)	(436.2847, 436.6370)
Titik 54	(504.1715, 406.4149)	(435.8807, 475.7003)

Langkah selanjutnya adalah membuat titik dunia, yaitu titik representasi sesungguhnya, dengan titik sudut paling kiri memiliki koordinat (0,0). Jarak antar satu titik pola planar ke titik disampingnya adalah 100 mm, sehingga

setiap menambah satu titik dunia kesamping maupun ke bawah, ditambah 100. Dimisalkan **titikDunia** sebagai titik dunia, berikut titik dunianya jika disusun dalam matriks,

$$\begin{bmatrix} (0,0) & (0,100) & (0,200) & (0,300) & (0,400) & (0,500) \\ (100,0) & (100,100) & (100,200) & (100,300) & (100,400) & (100,500) \\ (200,0) & (200,100) & (200,200) & (200,300) & (200,400) & (200,500) \\ (300,0) & (300,100) & (300,200) & (300,300) & (300,400) & (300,500) \\ (400,0) & (400,100) & (400,200) & (400,300) & (400,400) & (400,500) \\ (500,0) & (500,100) & (500,200) & (500,300) & (500,400) & (500,500) \\ (600,0) & (600,100) & (600,200) & (600,300) & (600,400) & (600,500) \\ (700,0) & (700,100) & (700,200) & (700,300) & (700,400) & (700,500) \\ (800,0) & (800,100) & (800,200) & (800,300) & (800,400) & (800,500) \end{bmatrix}^T$$

Titik citra dan titik dunia yang didapatkan digunakan untuk menyusun matriks **X** seperti pada persamaan (4.1). Sebagai langkah awal, titik citra yang digunakan adalah citra pola kiri sehingga matriks homografi yang didapatkan adalah matriks homografi untuk pola planar kiri. Berikut matriks **X**,

$$\mathbf{X} = \begin{bmatrix} 504 & 406 & 1 & 0 & 0 & 0 & 0 & 0 \\ 505 & 367 & 1 & 0 & 0 & 0 & 0 & 0 \\ 505 & 327 & 1 & 0 & 0 & 0 & 0 & 0 \\ 506 & 288 & 1 & 0 & 0 & 0 & 0 & 0 \\ 506 & 249 & 1 & 0 & 0 & 0 & 0 & 0 \\ 507 & 211 & 1 & 0 & 0 & 0 & 0 & 0 \\ 465 & 405 & 1 & 0 & 0 & 0 & -46474 & -40514 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 195 & 280 & 1 & 0 & 0 & 0 & -156279 & -223764 \\ 197 & 241 & 1 & 0 & 0 & 0 & -157555 & -192859 \\ 198 & 203 & 1 & 0 & 0 & 0 & -158632 & -162385 \\ 0 & 0 & 0 & 504 & 406 & 1 & 0 & 0 \\ 0 & 0 & 0 & 505 & 367 & 1 & -50480 & -36685 \\ 0 & 0 & 0 & 505 & 327 & 1 & -101050 & -65464 \\ 0 & 0 & 0 & 506 & 288 & 1 & -151768 & -86421 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 192 & 357 & 1 & -19202 & -35727 \\ 0 & 0 & 0 & 194 & 318 & 1 & -38800 & -63662 \\ 0 & 0 & 0 & 195 & 280 & 1 & -58605 & -83912 \\ 0 & 0 & 0 & 197 & 241 & 1 & -78778 & -96429 \\ 0 & 0 & 0 & 198 & 203 & 1 & -99145 & -101491 \end{bmatrix}$$

Matriks **X** kemudian disusun matriks **U** seperti pada persamaan (4.2) sebagai berikut

$$\mathbf{U} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \\ 100 \\ 100 \\ \vdots \\ 200 \\ 300 \\ 400 \\ 500 \end{bmatrix}$$

Matriks \mathbf{U} dan \mathbf{X} digunakan untuk menghitung matriks homografi menggunakan persamaan (2.2-2.4). Langkah awal dalam menghitung matriks homografi adalah dengan menghitung matriks \mathbf{tvec} seperti pada persamaan (2.2), sehingga didapatkan matriks \mathbf{tvec} ,

$$\mathbf{tvec} = \begin{bmatrix} -2.7 \\ -0 \\ 1354 \\ 0 \\ -2.6 \\ 1032 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Matriks \mathbf{T} yang dihasilkan adalah berikut ini,

$$\mathbf{T} = \begin{bmatrix} \mathbf{tvec}(1, 1) & \mathbf{tvec}(4, 1) & \mathbf{tvec}(7, 1) \\ \mathbf{tvec}(2, 1) & \mathbf{tvec}(5, 1) & \mathbf{tvec}(8, 1) \\ \mathbf{tvec}(3, 1) & \mathbf{tvec}(6, 1) & \mathbf{tvec}(9, 1) \end{bmatrix}^{-1} = \begin{bmatrix} -0.3714 & -0.0075 & 0.0000 \\ 0.0219 & -0.3651 & 0.0000 \\ 480.1461 & 386.8651 & 0.9523 \end{bmatrix}$$

Setiap elemen \mathbf{T} dibagi dengan $\mathbf{T}(3, 3)$ yaitu 0.9523, sehingga didapatkan matriks sebagai berikut,

$$\begin{bmatrix} -0.3900 & -0.0079 & 0.0000 \\ 0.0230 & -0.3834 & 0.0000 \\ 504.1766 & 406.2271 & 1.0000 \end{bmatrix}$$

Matriks homografi dari pola planar kiri didapatkan dengan mentranpose matriks \mathbf{T} yang setiap elemennya telah dibagi dengan $\mathbf{T}(3,3)$. Matriks homografi yang didapatkan adalah sebagai berikut,

$$\mathbf{H} = \begin{bmatrix} -0.3900 & 0.0230 & 504.1766 \\ -0.0079 & -0.3834 & 406.2271 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

Dengan cara yang sama, maka didapatkan homografi dari pola planar kanan adalah

$$\mathbf{H} = \begin{bmatrix} -0.3820 & 0.0352 & 435.7357 \\ -0.0118 & -0.3598 & 475.3133 \\ 0.0000 & 0.0001 & 1.0000 \end{bmatrix}$$

Matriks \mathbf{H} yang telah didapatkan baik pada pola planar kiri maupun kanan, digunakan untuk menghitung matriks \mathbf{V} seperti pada persamaan (4.4). Matriks \mathbf{V} dihasilkan sebagai berikut,

$$\mathbf{V} = \begin{bmatrix} -0.0090 & 0.1493 & 0.0030 & -0 & -0 & 0 \\ 0.1516 & 0.0238 & -0.1469 & -0 & 0 & -0 \\ -0.0134 & 0.1370 & 0.0042 & -0 & -0 & 0 \\ 0.1447 & 0.0343 & -0.1293 & -0 & 0 & -0 \end{bmatrix}$$

Matriks \mathbf{V} didekomposisikan dengan metode SVD, dan dimisalkan \mathbf{D} yang merupakan singular kanan dari \mathbf{V} . Berikut nilai dari \mathbf{D} ,

$$\mathbf{D} = \begin{bmatrix} 0.7139 & -0.1721 & -0.6787 & 0.0007 & -0.0018 & 0 \\ 0.2046 & 0.9783 & -0.0329 & 0.0001 & 0 & 0 \\ -0.6697 & 0.1154 & -0.7336 & 0.0006 & -0.0017 & 0 \\ -0 & -0.0001 & -0.0009 & 0.7557 & 0.6549 & -0.0001 \\ 0.0002 & -0 & -0.0025 & -0.6549 & 0.7557 & 0 \\ -0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Kolom terakhir dari matriks \mathbf{D} disusun seperti pada persamaan (2.5), sehingga didapatkan persamaan \mathbf{B} seperti berikut ini,

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & -0.0001 \\ 0 & 0 & 0 \\ -0 & 0 & 1 \end{bmatrix}$$

Pada fase ini parameter instrinsik dapat dihitung menggunakan persamaan (2.13-2.14), kemudian dihitung parameter ekstrinsik. Parameter-parameter yang dihasilkan, kemudian dioptimalkan agar errornya kecil menggunakan persamaan (4.7), berikut parameter intrinsik yang dihasilkan,

$$\mathbf{A} = \begin{bmatrix} 1010 & 0 & 338 \\ 0 & 1013 & 284 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Parameter ekstrinsi yang dihasilkan setelah proses optimasi parameter ditampilkan pada \mathbf{R}_{kl} , \mathbf{R}_{kk} dan \mathbf{t} berikut ini,

$$\mathbf{R}_{kl} = \begin{bmatrix} -0.9993 & -0.0273 & 0.0236 \\ 0.0293 & -0.9959 & 0.0855 \\ 0.0212 & 0.0861 & 0.9961 \end{bmatrix} \quad (5.2)$$

$$\mathbf{R}_{kk} = \begin{bmatrix} -0.9994 & -0.0322 & 0.0115 \\ 0.0337 & -0.9854 & 0.1666 \\ 0.0060 & 0.1669 & 0.9860 \end{bmatrix} \quad (5.3)$$

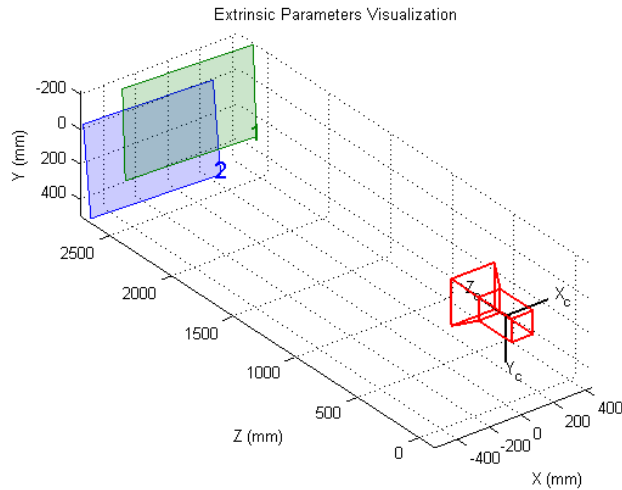
$$\mathbf{t} = \begin{bmatrix} 423 & 311 & 2573 \\ 256 & 499 & 2644 \end{bmatrix} \quad (5.4)$$

\mathbf{R}_{kl} , \mathbf{R}_{kk} adalah matriks rotasi pada kamera kiri dan kanan. \mathbf{t} adalah matriks translasi. Pengoptimalan parameter-parameter yang dihasilkan juga menghasilkan koefisien ditorsi, berikut koefisien distorsinya,

$$k_1 = 0.1104 \quad (5.5)$$

$$k_2 = -0.8901 \quad (5.6)$$

Parameter ekstrinsik yang dihasilkan, dapat divisualisasikan sebagai pada Gambar 5.5,



Gambar 5.5: Visualisasi parameter ekstrinsik

Uji coba untuk mendapatkan hasil kalibrasi kamera zhang yang bagus, dilakukan dengan melakukan kombinasi baik dari segi letak maupun jumlah

pada papan pola planar yang digunakan. Tujuan kombinasi tersebut untuk menguji kualitas parameter-parameter ekstrinsik dan intrinsik, uji kualitas itu berupa penghilangan distorsi citra. Hasil yang terbaik akan di-uji coba untuk melakukan estimasi jarak antara kamera dengan obyek. Akurasi estimasi bisa menjadi indikator kualitas hasil kalibrasi kamera.

Dalam kalibrasi kamera Zhang pada kamera stereo, minimal digunakan empat citra pola planar, yaitu dua citra dari kamera kanan dan dua dari kamera kiri. Hal ini disebabkan karena dalam kalibrasi kamera Zhang minimal digunakan dua citra pola planar. Kombinasi yang digunakan dari segi letak pola planar adalah kombinasi pola planar yang menghadap kamera dan pola planar yang miring dari kamera.

5.3.1 Uji Coba Kamera Sejajar dengan Obyek

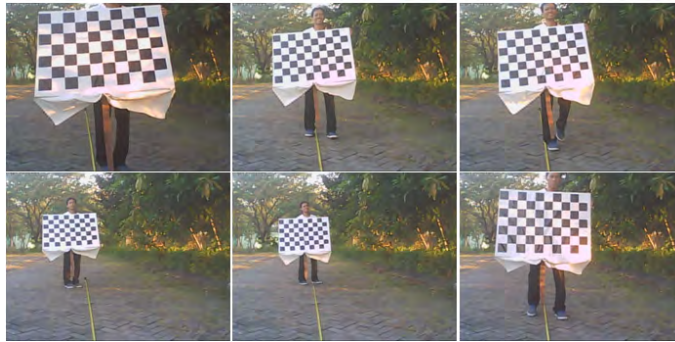
Uji coba dilakukan pada beberapa interval jarak, karena estimasi jarak adalah indikator kedua dalam menguji hasil kalibrasi kamera. Pengambilan data yang pertama dilakukan pada jarak 420 cm sampai 740 cm pada posisi kamera sejajar dengan obyek. Kombinasi citra yang digunakan untuk posisi menghadap kamera berjumlah 18 citra, Gambar 5.6 sampai Gambar 5.8 berikut adalah pola planar pada posisi menghadap kamera dari kamera kiri.



Gambar 5.6: Pola planar satu sampai enam pada kamera kiri dengan posisi menghadap kamera



Gambar 5.7: Pola planar tujuh sampai 12 pada kamera kiri dengan posisi menghadap kamera



Gambar 5.8: Pola planar 13 sampai 18 pada kamera kiri dengan posisi menghadap kamera

Kombinasi citra yang digunakan untuk posisi menghadap kamera pada kamera kanan juga berjumlah 18 citra, Gambar 5.9 sampai Gambar 5.11 berikut adalah pola planar pada posisi menghadap kamera dari kamera kanan.



Gambar 5.9: Pola planar satu sampai enam pada kamera kanan dengan posisi menghadap kamera



Gambar 5.10: Pola planar tujuh sampai 12 pada kamera kanan dengan posisi menghadap kamera









Gambar 5.11: Pola planar 13 sampai 18 pada kamera kanan dengan posisi menghadap kamera






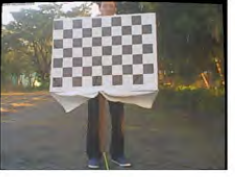
Pola planar kamera kiri dan kanan, dikombinasikan dari segi jumlah yaitu menggunakan empat sampai 36 citra pola planar untuk melakukan kalibrasi kamera. Empat citra menjadi jumlah pola planar terkecil karena dari empat citra tersebut dua dari kamera kiri dan dua dari kamera kanan. Gambar 5.12 sampai Gambar 5.17 adalah hasil penghilangan distorsi citra menggunakan kombinasi pola planar dari empat sampai 36,

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
4		
6		
8		

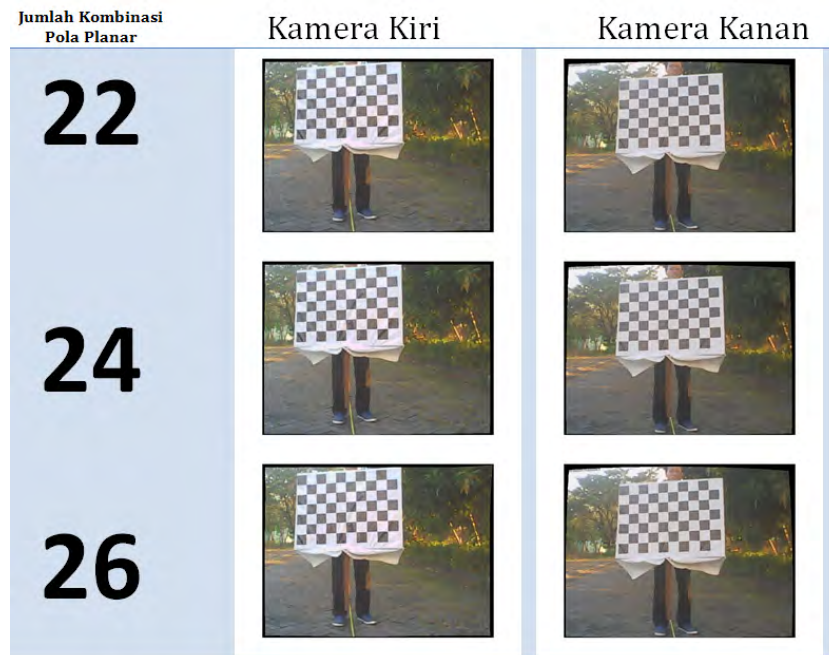
Gambar 5.12: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar empat sampai delapan dengan posisi menghadap kamera

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
10		
12		
14		

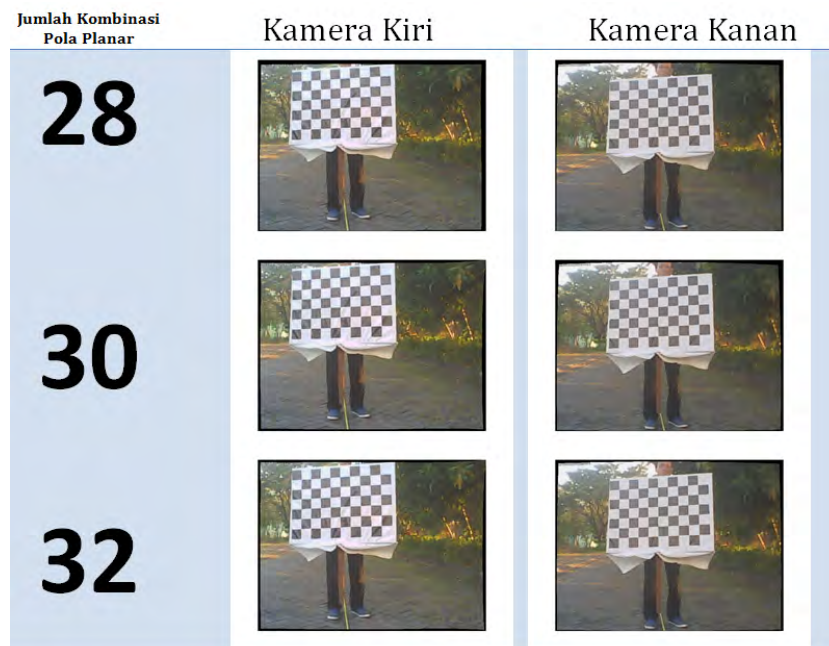
Gambar 5.13: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar sepuluh sampai 14 dengan posisi menghadap kamera

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
16		
18		
20		





Gambar 5.14: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 16 sampai 20 dengan posisi menghadap kamera



Gambar 5.15: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 22 sampai 26 dengan posisi menghadap kamera



Gambar 5.16: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 28 sampai 32 dengan posisi menghadap kamera

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
34		
36		

Gambar 5.17: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 34 sampai 36 dengan posisi menghadap kamera

Kombinasi citra dengan posisi miring dari kamera kiri berjumlah 18 citra, Gambar 5.18 sampai Gambar 5.19 berikut adalah contoh 12 pola planar pada posisi miring yang digunakan dari kamera kiri,



Gambar 5.18: Pola planar satu sampai enam pada kamera kiri dengan posisi miring dari kamera



Gambar 5.19: Pola planar tujuh sampai 12 pada kamera kiri dengan posisi miring dari kamera

Kombinasi citra yang digunakan untuk posisi miring dari kamera pada kamera kanan juga berjumlah 18 citra, Gambar 5.20 sampai Gambar 5.21 berikut adalah contoh 12 pola planar pada posisi miring yang digunakan dari kamera kanan,









Gambar 5.20: Pola planar satu sampai enam pada kamera kanan dengan posisi miring dari kamera









Gambar 5.21: Pola planar tujuh sampai 12 pada kamera kanan dengan posisi miring dari kamera

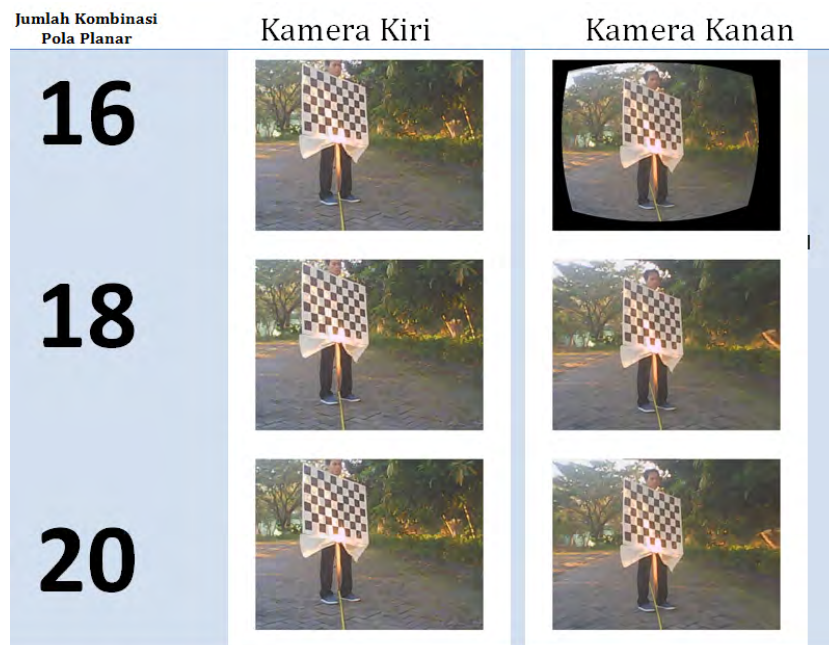
Pola planar posisi miring adalah posisi pola panar yang miring ke samping di depan kamera. Pola planar posisi miring dibutuhkan agar memberi orientasi lain dari matriks homografi, karena apabila citra terlalu mirip yaitu adakalanya hasil kalibrasi kamera tidak begitu bagus. Pengambilan posisi pola planar miring di sini dilakukan pada jarak dekat dari kamera dan jarak jauh dari kamera untuk memberikan orientasi yang berbeda ketika posisi miring di dekat dan jauh dari kamera. Pola planar kamera kiri dan kanan, dikombinasikan dari segi jumlah yaitu menggunakan empat sampai 36 citra pola planar untuk melakukan kalibrasi kamera. Gambar 5.22 sampai Gambar 5.27 adalah hasil penghilangan distorsi citra menggunakan kombinasi pola planar dari empat sampai 36,

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
4		
6		
8		

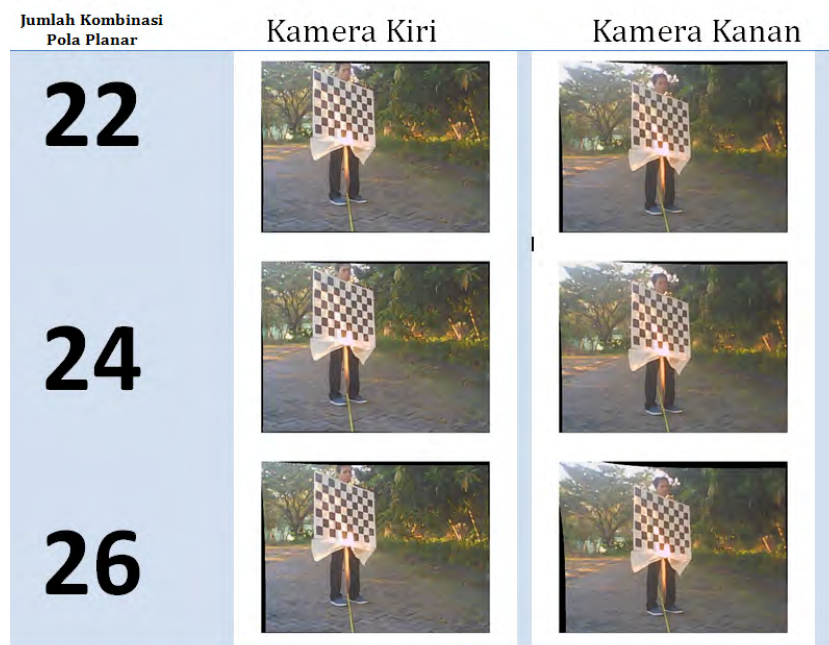
Gambar 5.22: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar empat sampai delapan dengan posisi miring dari kamera

Jumlah Kombinasi Pola Planar	Kamera Kiri	Kamera Kanan
10		
12		
14		

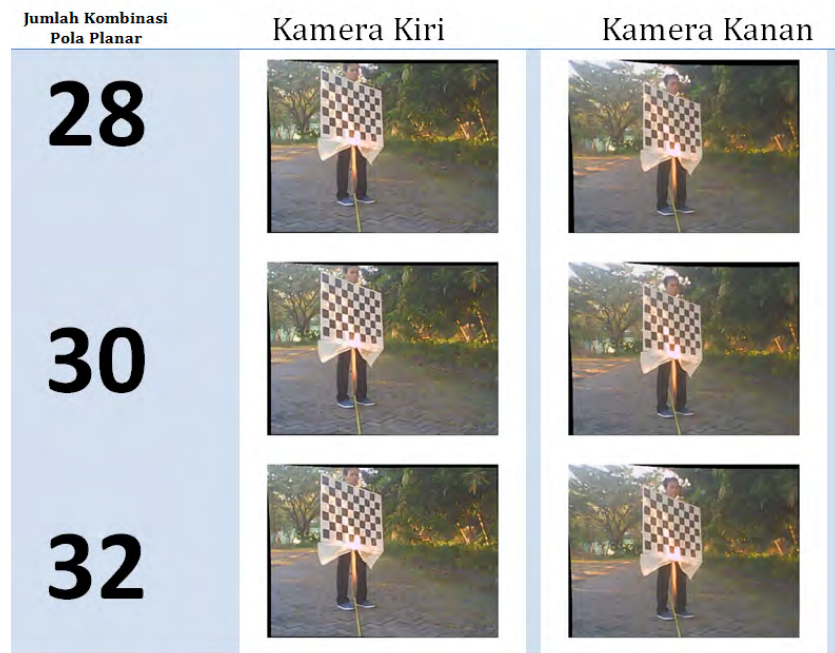
Gambar 5.23: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar sepuluh sampai 14 dengan posisi miring dari kamera



Gambar 5.24: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 16 sampai 20 dengan posisi miring dari kamera



Gambar 5.25: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 22 sampai 26 dengan posisi miring dari kamera

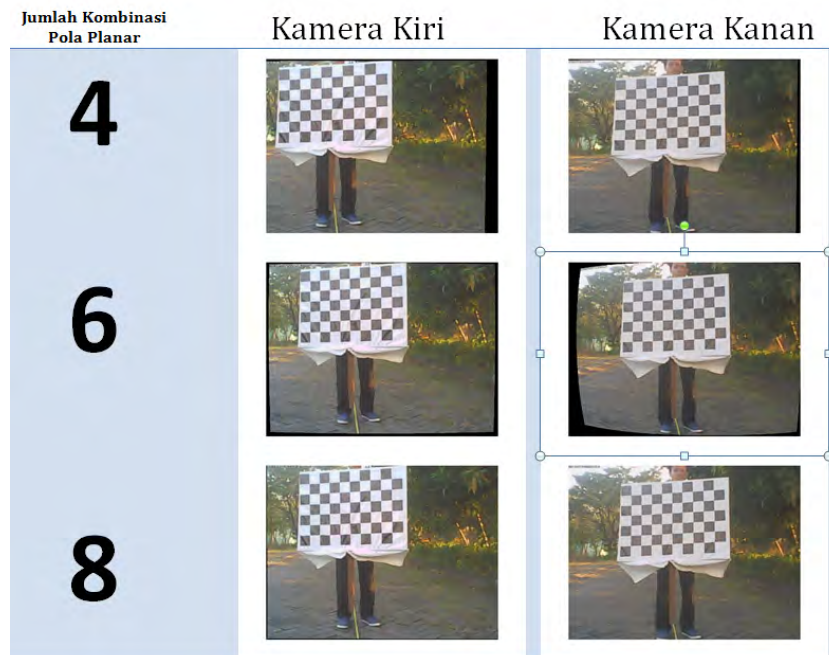


Gambar 5.26: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 28 sampai 32 dengan posisi miring dari kamera

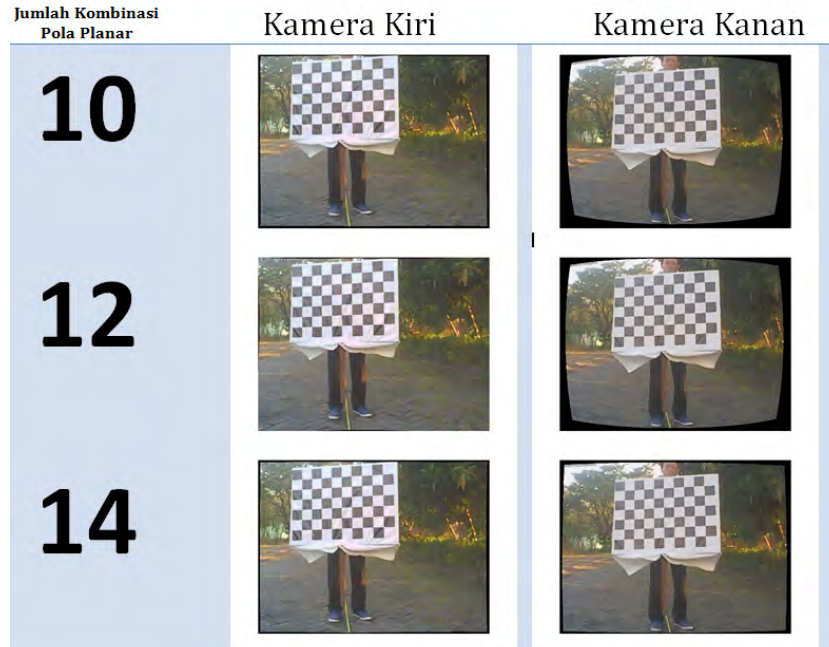


Gambar 5.27: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 34 sampai 36 dengan posisi miring dari kamera

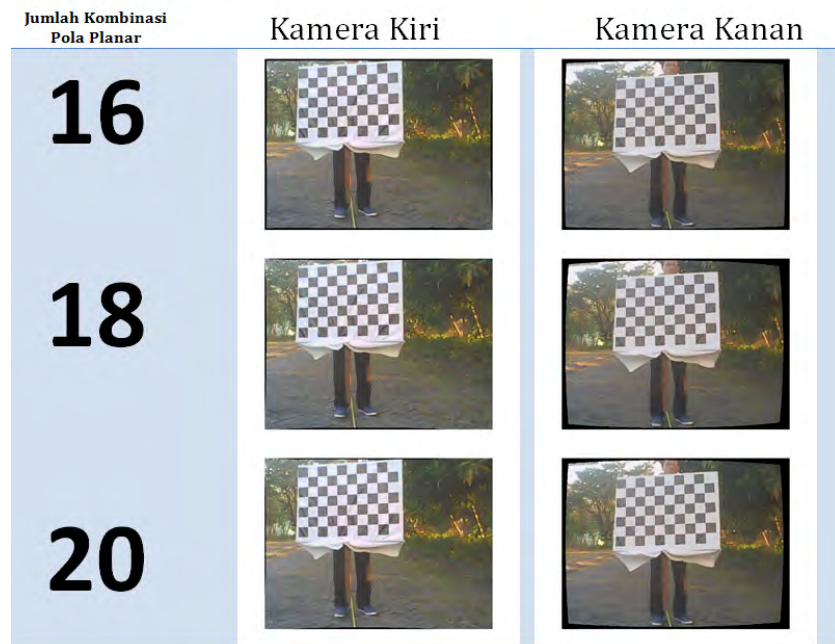
Gambar 5.28 sampai Gambar 5.33 adalah hasil penghilangan distorsi citra menggunakan kombinasi pola planar gabungan posisi miring dan menghadap kamera dengan kombinasi jumlah pola planar empat sampai 36,



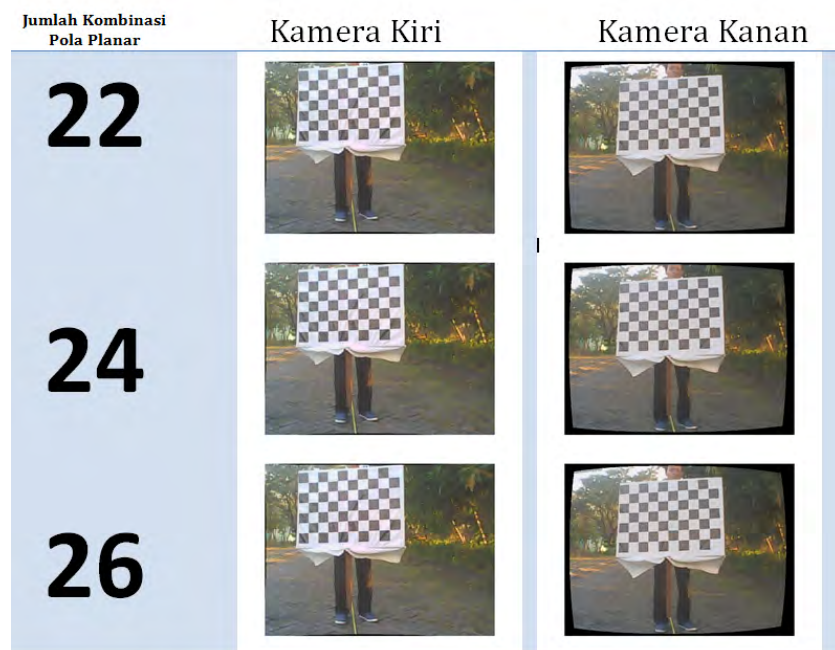
Gambar 5.28: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar empat sampai delapan dengan posisi gabungan miring dan menghadap dari kamera



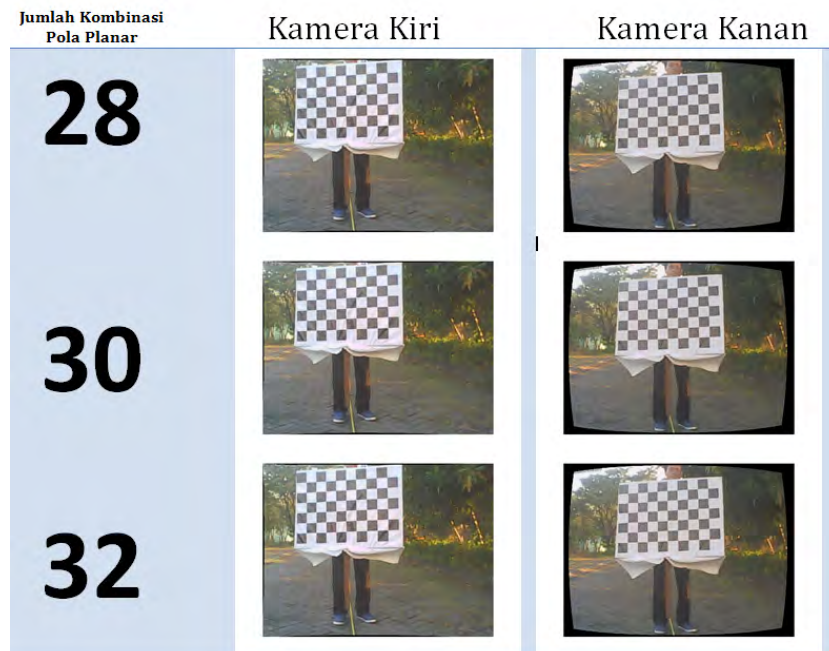
Gambar 5.29: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar sepuluh sampai 14 dengan posisi gabungan miring dan menghadap dari kamera



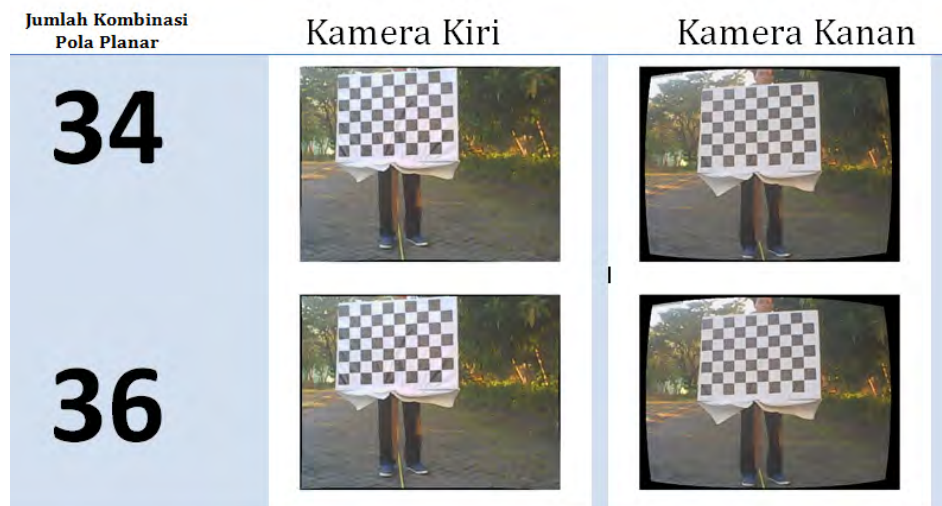
Gambar 5.30: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 16 sampai 20 dengan posisi gabungan miring dan menghadap dari kamera



Gambar 5.31: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 22 sampai 26 dengan posisi gabungan miring dan menghadap dari kamera



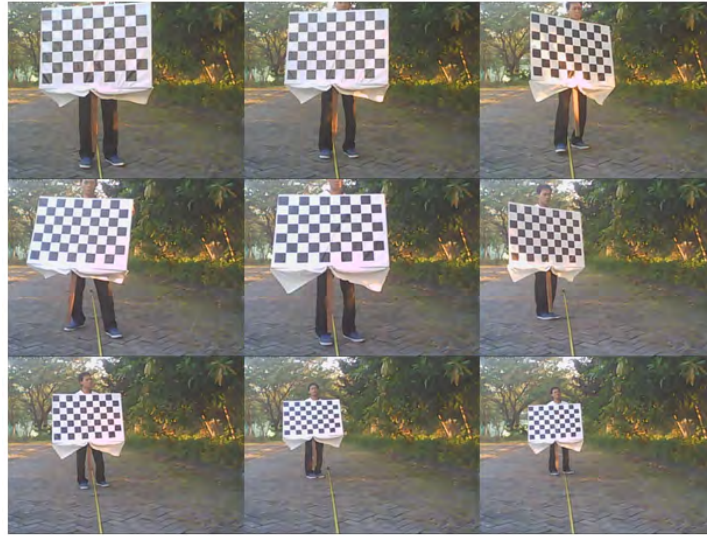
Gambar 5.32: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 28 sampai 32 dengan posisi gabungan miring dan menghadap dari kamera



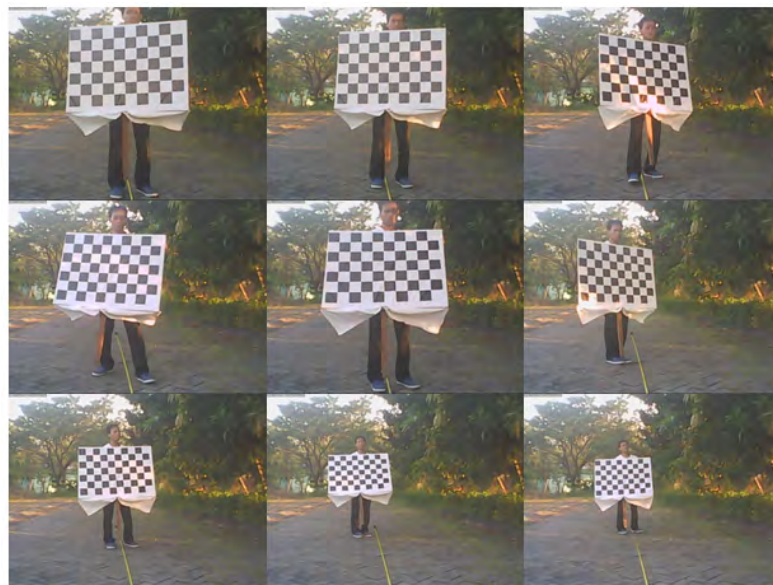
Gambar 5.33: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 34 sampai 36 dengan posisi gabungan miring dan menghadap dari kamera

Melihat hasil yang didapat di atas, dapat disimpulkan bahwa hasil terbaik adalah saat pola planar menghadap ke kamera dengan kombinasi jumlah tidak begitu berpengaruh. Uji coba kedua adalah dengan melakukan estimasi jarak, maka dipilih salah satu hasil penghilangan distorsi citra yang terbaik. Citra pola planar yang digunakan pada data ini ditampilkan pada Gambar

5.34 dan Gambar 5.35.



Gambar 5.34: Citra pola planar kiri



Gambar 5.35: Citra pola planar kanan

Setiap pola planar yang digunakan harus dideteksi titik-titik yang ada didalamnya. Karena koordinat pola planar yang terdeteksi akan digunakan dalam perhitungan homografi antar citra. Pola planar yang digunakan dalam penelitian ini digunakan pola planar berukuran 7×10 , sehingga dimiliki 54 titik pada pola planar. Terlebih dahulu ditunjukkan dahulu deteksi titik pola planar pertama pada semua pola yang digunakan, baik dari kamera kiri maupun kamera kanan, berikut hasilnya seperti yang ditunjukkan pada Gambar 5.36 dan Gambar 5.37 berikut ini,

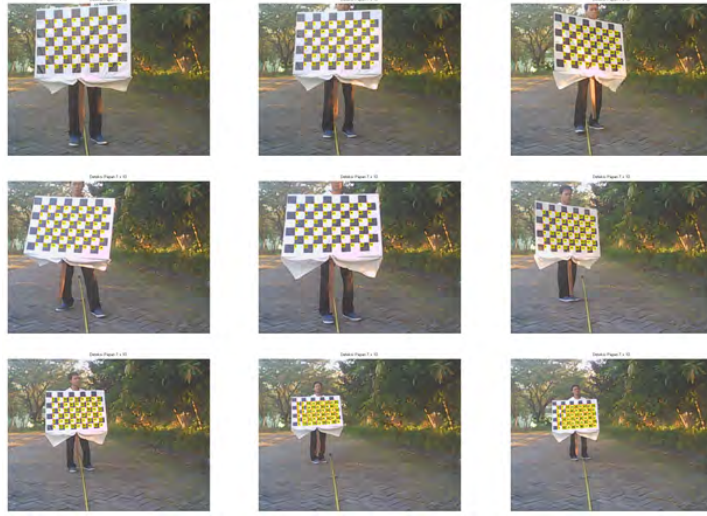


Gambar 5.36: Deteksi titik pada pola planar, pada citra dari kamera kiri



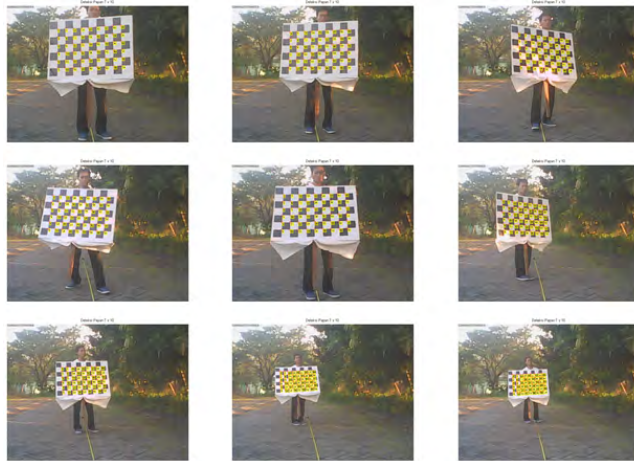
Gambar 5.37: Deteksi titik pada pola planar, pada citra dari kamera kanan

Gambar 5.36 dan Gambar 5.37 menunjukkan titik pola planar yang terdeteksi berjumlah 54 titik, jumlah itu sesuai dengan jumlah titik pola planar sesungguhnya. Pola planar yang digunakan pada citra yang lainnya harus sama dengan pola planar yang digunakan pada pola planar pertama. Apabila pola planar yang digunakan berbeda baik dari segi jumlah titik maupun ukuran pola planar, maka estimasi pola planar bisa menghasilkan hasil yang tidak sesuai atau bahkan error pada perhitungannya disebabkan perbedaan titik pada pola planar yang digunakan. Berikut ini pada Gambar 5.38 ditunjukkan deteksi semua pola planar yang dihasilkan oleh kamera kiri,



Gambar 5.38: Deteksi semua pola planar pada citra dari kamera kiri

Deteksi semua pola planar yang dihasilkan oleh kamera kanan ditunjukkan pada Gambar 5.39 berikut ini,



Gambar 5.39: Deteksi semua pola planar pada citra dari kamera kanan

Parameter intrinsik pada kamera kiri yang dihasilkan adalah,

$$\mathbf{A}_{ki} = \begin{bmatrix} 2298 & 0 & 857 \\ 0 & 2114 & -1492 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

Parameter ekstrinsiknya pada kamera kiri adalah

$$\mathbf{R}_{ki}(:, :, 1) = \begin{bmatrix} 0.9645 & -0.0417 & 0.2608 \\ 0.2386 & 0.5609 & -0.7927 \\ -0.1132 & 0.8268 & 0.5510 \end{bmatrix} \quad (5.8)$$

$$\mathbf{R}_{ki}(:, :, 2) = \begin{bmatrix} 0.7906 & 0.3918 & 0.4707 \\ 0.0993 & 0.6764 & -0.7298 \\ -0.6043 & 0.6237 & 0.4959 \end{bmatrix} \quad (5.9)$$

$$\mathbf{R}_{ki}(:, :, 3) = \begin{bmatrix} 0.9886 & -0.1086 & 0.1047 \\ 0.1486 & 0.5825 & -0.7992 \\ 0.0258 & 0.8056 & 0.5919 \end{bmatrix} \quad (5.10)$$

$$\mathbf{R}_{ki}(:, :, 4) = \begin{bmatrix} 0.9925 & -0.0815 & 0.0910 \\ 0.1195 & 0.8036 & -0.5831 \\ -0.0256 & 0.5896 & 0.8073 \end{bmatrix} \quad (5.11)$$

$$\mathbf{R}_{ki}(:, :, 5) = \begin{bmatrix} 0.9811 & -0.0699 & 0.1804 \\ 0.1663 & 0.7809 & -0.6021 \\ -0.0987 & 0.6207 & 0.7778 \end{bmatrix} \quad (5.12)$$

$$\mathbf{R}_{ki}(:, :, 6) = \begin{bmatrix} 0.9721 & -0.0395 & 0.2314 \\ 0.1838 & 0.7411 & -0.6458 \\ -0.1459 & 0.6702 & 0.7276 \end{bmatrix} \quad (5.13)$$

$$\mathbf{R}_{ki}(:, :, 7) = \begin{bmatrix} 0.9548 & 0.1221 & 0.2711 \\ 0.1365 & 0.6301 & -0.7644 \\ -0.2641 & 0.7668 & 0.5850 \end{bmatrix} \quad (5.14)$$

$$\mathbf{R}_{ki}(:, :, 8) = \begin{bmatrix} 0.9653 & -0.0481 & 0.2565 \\ 0.2068 & 0.7405 & -0.6394 \\ -0.1592 & 0.6703 & 0.7248 \end{bmatrix} \quad (5.15)$$

$$\mathbf{R}_{ki}(:, :, 9) = \begin{bmatrix} 0.9731 & -0.0906 & 0.2118 \\ 0.2056 & 0.7562 & -0.6212 \\ -0.1039 & 0.6480 & 0.7545 \end{bmatrix} \quad (5.16)$$

$$\mathbf{t}_{ki} = \begin{bmatrix} -2606 & 7930 & 9230 \\ -2745 & 7785 & 9270 \\ -2207 & 6057 & 7144 \\ -2537 & 7478 & 8940 \\ -2922 & 8470 & 9932 \\ -3146 & 9332 & 10891 \\ -2952 & 9241 & 10986 \\ -2784 & 8798 & 10306 \\ -3164 & 9294 & 10830 \end{bmatrix} \quad (5.17)$$

\mathbf{R}_{ki} adalah matriks rotasi pada kamera kiri. $\mathbf{R}_{ki}(:, :, 1)$ adalah matriks rotasi kiri pada citra pola planar pertama dari kamera kiri kemudian $\mathbf{R}_{ki}(:, :, 2)$ adalah matriks rotasi pada pola planar kedua dari kamera kiri, begitu pula selanjutnya. \mathbf{t}_{ki} adalah matriks translasi pada kamera kiri. Koefisien distorsi pada kamera kiri adalah

$$k_1 = -0.1530 \quad (5.18)$$

$$k_2 = 0.0196 \quad (5.19)$$

Parameter intrinsik pada kamera kanan yang dihasilkan adalah,

$$\mathbf{A}_{ka} = \begin{bmatrix} 2672 & 0 & 676 \\ 0 & 2502 & -661 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.20)$$

Parameter ekstrinsiknya pada kamera kanan adalah

$$\mathbf{R}_{ka}(:, :, 1) = \begin{bmatrix} 0.9792 & -0.1143 & 0.1674 \\ 0.1858 & 0.8362 & -0.5160 \\ -0.0810 & 0.5364 & 0.8401 \end{bmatrix} \quad (5.21)$$

$$\mathbf{R}_{ka}(:, :, 2) = \begin{bmatrix} 0.8174 & 0.1950 & 0.5420 \\ 0.0503 & 0.9131 & -0.4045 \\ -0.5738 & 0.3579 & 0.7366 \end{bmatrix} \quad (5.22)$$

$$\mathbf{R}_{ka}(:, :, 3) = \begin{bmatrix} 0.9936 & -0.1130 & -0.0035 \\ 0.0955 & 0.8559 & -0.5082 \\ 0.0604 & 0.5046 & 0.8612 \end{bmatrix} \quad (5.23)$$

$$\mathbf{R}_{ka}(:, :, 4) = \begin{bmatrix} 0.9966 & -0.0826 & -0.0055 \\ 0.0793 & 0.9721 & -0.2207 \\ 0.0236 & 0.2195 & 0.9753 \end{bmatrix} \quad (5.24)$$

$$\mathbf{R}_{ka}(:, :, 5) = \begin{bmatrix} 0.9908 & -0.1077 & 0.0816 \\ 0.1250 & 0.9602 & -0.2498 \\ -0.0514 & 0.2577 & 0.9649 \end{bmatrix} \quad (5.25)$$

$$\mathbf{R}_{ka}(:, :, 6) = \begin{bmatrix} 0.9849 & -0.1003 & 0.1409 \\ 0.1398 & 0.9415 & -0.3066 \\ -0.1019 & 0.3217 & 0.9413 \end{bmatrix} \quad (5.26)$$

$$\mathbf{R}_{ka}(:, :, 7) = \begin{bmatrix} 0.9696 & 0.0316 & 0.2426 \\ 0.0854 & 0.8855 & -0.4568 \\ -0.2293 & 0.4636 & 0.8559 \end{bmatrix} \quad (5.27)$$

$$\mathbf{R}_{ka}(:, :, 8) = \begin{bmatrix} 0.9798 & -0.1185 & 0.1609 \\ 0.1632 & 0.9391 & -0.3023 \\ -0.1153 & 0.3225 & 0.9395 \end{bmatrix} \quad (5.28)$$

$$\mathbf{R}_{ka}(:, :, 9) = \begin{bmatrix} 0.9849 & -0.1394 & 0.1026 \\ 0.1630 & 0.9463 & -0.2793 \\ -0.0581 & 0.2918 & 0.9547 \end{bmatrix} \quad (5.29)$$

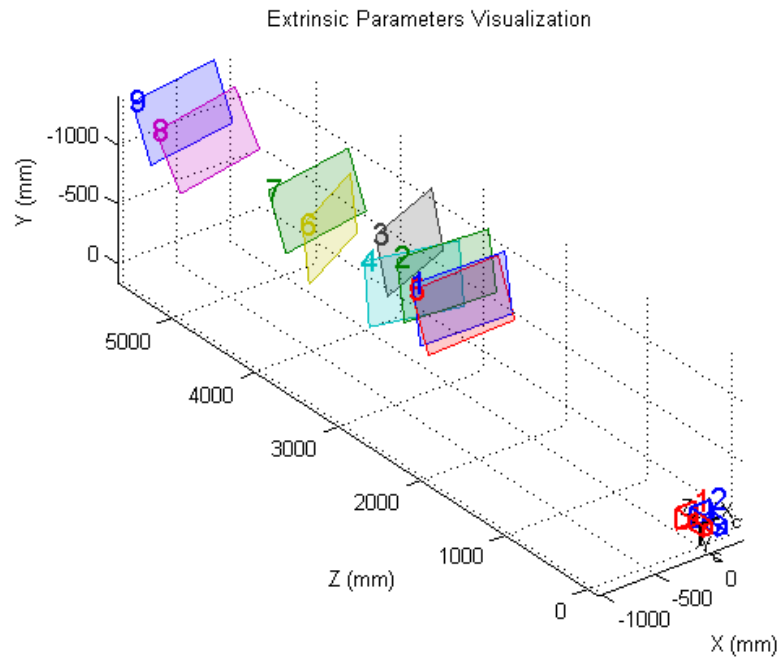
$$\mathbf{t}_{ka} = \begin{bmatrix} -2092 & 3462 & 11567 \\ -2228 & 3309 & 11553 \\ -1820 & 2584 & 8888 \\ -2040 & 3164 & 11117 \\ -2365 & 3670 & 12442 \\ -2530 & 4073 & 13676 \\ -2331 & 3957 & 13716 \\ -2205 & 3825 & 12907 \\ -2552 & 4062 & 13606 \end{bmatrix} \quad (5.30)$$

\mathbf{R}_{ka} adalah matriks rotasi pada kamera kanan. \mathbf{t}_{ka} adalah matriks translasi pada kamera kiri. Koefisien distorsi pada kamera kanan adalah

$$k_1 = -0.1737 \quad (5.31)$$

$$k_2 = 0.1915 \quad (5.32)$$

Parameter ekstrinsik yang dihasilkan adalah representasi dari keadaan kamera ketika kamera mengambil citra pola planar. Setiap pergantian posisi maupun jarak pada pola planar tercermin dalam setiap parameter ekstrinsik baik itu dalam posisi pola planar menghadap kamera maupun ketika pola planar dalam keadaan miring. Bahkan pola planar yang dekat dan jauh dari kamera ketika pengambilan datanya juga tercermin dalam parameter ekstrinsik. Oleh karena itu parameter ekstrinsik yang meliputi rotasi dan translasi dapat divisualisasikan dalam bentuk grafik untuk menggambarkan posisi pola planar ketika pengambilan data dilakukan, Gambar 5.40 berikut ini adalah visualisasi dari parameter ekstrinsik yang dihasilkan dari kalibrasi kamera zhang,



Gambar 5.40: Visualisasi parameter ekstrinsik pada pola planar

Gambar 5.41 dan Gambar 5.42 berikut ini adalah hasil penghilangan distorsi menggunakan koefisien distorsi yang digunakan,



Gambar 5.41: Hasil penghilangan distorsi pada kamera kiri



Gambar 5.42: Hasil penghilangan distorsi pada kamera kanan

Citra yang diambil, dihilangkan distorsinya kemudian dideteksi koordinat pojok kiri obyek yang akan diestimasi jaraknya. Uji coba dilakukan empat kali untuk melihat kehandalan sistem dalam mengestimasi jarak. Cara untuk mendapatkan koordinat pojok kiri dari obyek, dilakukan dengan cara manual, yaitu dengan mendeteksi obyek kemudian diambil koordinat pojok kiri dari obyek. Data koordinat pojok kiri pada uji coba pertama ditampilkan pada Tabel 5.2 berikut ini,

Tabel 5.2: Jarak dan koordinat pojok kiri obyek citra kiri maupun kanan pada uji coba pertama dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Koordinat citra kiri	Koordinat citra kanan
420	(296,102)	(377,151)
440	(290,109)	(370,161)
460	(301,116)	(379,168)
480	(295,195)	(372,176)
500	(306,134)	(382,186)
520	(307,142)	(380,194)
540	(307,148)	(378,200)
560	(384,154)	(312,204)
580	(306,159)	(377,210)
600	(306,163)	(376,214)
620	(306,183)	(375,234)
640	(307,180)	(375,230)
660	(310,180)	(377,230)
680	(309,189)	(376,235)
700	(315,205)	(382,246)
720	(311,203)	(375,254)
740	(311,198)	(377,248)

Tabel 5.3 menunjukkan hasil dispariti dan faktor citra pada uji coba pertama,

Tabel 5.3: Faktor citra pada uji coba pertama dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Dispariti (piksel)	Faktor
420	81	34020
440	80	35200
460	78	35880
480	77	36960
500	76	38000
520	73	37960
540	71	38340
560	72	40320
580	71	41180
600	70	42000
620	69	42780
640	68	43520
660	67	44220
680	67	45560
700	67	46900
720	64	46080
740	66	48840

Faktor-faktor yang sudah didapatkan pada Tabel 5.3, didapatkan rata-rata faktor pada uji coba pertama adalah 41044.70588, dengan cara yang sama didapatkan rata-rata faktor pada uji coba kedua adalah 41964.71, uji coba ketiga adalah 4148.71 dan uji coba keempat adalah 41561.18. Persamaan (4.22) digunakan untuk melakukan estimasi jarak, sehingga jarak estimasi dan jarak yang sesungguhnya dibandingkan untuk mendapatkan error estimasi. Tabel 5.4 berikut ini, ditampilkan hasil perbandingan jarak sesungguhnya dengan jarak hasil estimasi pada keempat uji coba,

Tabel 5.4: Estimasi jarak dan presentase error dengan posisi kamera sejajar dengan obyek

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
420	1. 506.7247	0.2064
	2. 499.5798	0.1894
	3. 512.1568	0.2194
	4. 513.1009	0.2216

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
440	1. 513.0588 2. 518.0827 3. 525.1228 4. 506.8436	0.1660 0.1774 0.1934 0.1519
460	1. 526.2141 2. 538.0090 3. 531.8552 4. 532.8355	0.1439 0.1695 0.1562 0.1583
480	1. 533.0481 2. 499.5798 3. 538.7624 4. 546.8575	0.1105 0.0407 0.1224 0.1392
500	1. 540.0619 2. 552.1671 3. 568.2836 4. 569.3311	0.0801 0.1043 0.1365 0.1386
520	1. 562.2562 2. 574.8589 3. 545.8513 4. 546.8575	0.0812 0.1054 0.0497 0.0516
540	1. 578.0944 2. 574.8589 3. 568.2836 4. 561.6375	0.0705 0.0645 0.0523 0.0400
560	1. 570.0653 2. 582.8431 3. 584.2916 4. 577.2385	0.0179 0.0407 0.0433 0.0307
580	1. 578.0944 2. 582.8431 3. 592.6386 4. 593.7310	0.0032 0.0049 0.0217 0.0236
600	1. 586.3529 2. 599.4957 3. 584.2916 4. 569.3311	0.0227 0.0008 0.0261 0.0511
620	1. 594.8508 2. 599.4957 3. 576.1764 4. 593.7310	0.0405 0.0330 0.0706 0.0423
640	1. 603.5986 2. 617.1280 3. 592.6386 4. 611.1937	0.0568 0.0357 0.0740 0.0450

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
660	1. 612.6075	0.0718
	2. 599.4957	0.0916
	3. 601.2276	0.0890
	4. 602.3358	0.0873
680	1. 612.6075	0.0991
	2. 626.3388	0.0789
	3. 601.2276	0.1158
	4. 611.1937	0.1011
700	1. 612.6075	0.1248
	2. 567.0906	0.1898
	3. 638.2262	0.0882
	4. 611.1937	0.1268
720	1. 641.3235	0.1092
	2. 626.3388	0.1300
	3. 619.1747	0.1400
	4. 620.3160	0.1384
740	1. 621.8894	0.1596
	2. 645.6108	0.1275
	3. 619.1747	0.1632
	4. 629.7147	0.1490

Median dari rasio error relatif pada uji coba pertama adalah 0.0812, pada uji coba kedua adalah 0.0917, pada uji coba ketiga adalah 0.089 dan yang keempat adalah 0.1012. Keempat uji coba yang sudah dilakukan, didapatkan kesimpulan bahwa sistem yang digunakan cukup stabil dalam estimasi jarak.

Perancangan GMM pada data ini menggunakan inisialisasi nilai varian adalah 0.00009, sedangkan laju pelatihan adalah 0.001, kemudian banyak model *background* yang digunakan adalah tujuh. Morfologi citra yang digunakan adalah filtering, erosi dan filling. Langkah terakhir adalah dengan melakukan estimasi jarak pada obyek bergerak. Konsep yang digunakan pada estimasi jarak pada obyek bergerak dalam kasus ini adalah dengan melakukan deteksi obyek pada kedua video yang dihasilkan baik pada kemra kanan maupun kiri. Koordinat pojok kiri obyek yang terdeteksi pada kedua video itulah yang digunakan untuk menghitung diparity agar bisa di-estimasi jaraknya antara obyek dengan kamera. Keakuratan hasil estimasi jarak akan terganggu apabila pojok kiri yang terdeteksi pada kedua video tidak tepat pada titik yang sama. Simulasi estimasi jarak pada obyek bergerak ditunjukkan pada Gambar 5.43 berikut ini,



Gambar 5.43: Deteksi obyek bergerak beserta estimasi jarak dari kamera ke obyek dengan posisi kamera sejajar dengan obyek

Pengambilan data yang kedua dilakukan pada jarak 750 cm sampai 1350 cm di depan kamera. Hampir sebagian besar data pada interval jarak ini adalah data pola planar yang menghadap ke kamera dan hanya satu data yang dalam keadaan miring dari kamera. Kombinasi jumlah dan posisi pola planar di data ini didapatkan hasil terbaik ketika dua posisi miring digunakan bersama dengan 18 pola menghadap kamera. Oleh karena itu dipilih kombinasi citra pada dua posisi miring dan 18 posisi menghadap kamera. Parameter intrinsik pada kamera kiri yang dihasilkan adalah,

$$\mathbf{A}_{ki} = \begin{bmatrix} 2324 & 0 & 627 \\ 0 & 2202 & -676 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.33)$$

Sedangkan parameter ekstrinsiknya pada kamera kiri adalah

$$\mathbf{R}_{ki}(:, :, 1) = \begin{bmatrix} 0.9884 & -0.1519 & 0.0073 \\ 0.1242 & 0.7783 & -0.6155 \\ 0.0879 & 0.6093 & 0.7881 \end{bmatrix} \quad (5.34)$$

$$\mathbf{R}_{ki}(:, :, 2) = \begin{bmatrix} 0.8438 & 0.2144 & 0.4919 \\ 0.0362 & 0.8919 & -0.4508 \\ -0.5354 & 0.3982 & 0.7449 \end{bmatrix} \quad (5.35)$$

$$\mathbf{R}_{ki}(:, :, 3) = \begin{bmatrix} 0.9883 & -0.1326 & 0.0755 \\ 0.1490 & 0.7311 & -0.6658 \\ 0.0331 & 0.6693 & 0.7423 \end{bmatrix} \quad (5.36)$$

$$\mathbf{R}_{ki}(:, :, 4) = \begin{bmatrix} 0.9855 & -0.1623 & 0.0490 \\ 0.1687 & 0.9091 & -0.3808 \\ 0.0172 & 0.3836 & 0.9233 \end{bmatrix} \quad (5.37)$$

$$\mathbf{R}_{ki}(:, :, 5) = \begin{bmatrix} 0.9831 & -0.0243 & 0.1817 \\ 0.1082 & 0.8769 & -0.4684 \\ -0.1479 & 0.4801 & 0.8647 \end{bmatrix} \quad (5.38)$$

$$\mathbf{R}_{ki}(:, :, 6) = \begin{bmatrix} 0.9861 & -0.0571 & 0.1560 \\ 0.1278 & 0.8610 & -0.4923 \\ -0.1062 & 0.5054 & 0.8563 \end{bmatrix} \quad (5.39)$$

$$\mathbf{R}_{ki}(:, :, 7) = \begin{bmatrix} 0.9815 & -0.0788 & 0.1744 \\ 0.1498 & 0.8833 & -0.4443 \\ -0.1190 & 0.4622 & 0.8788 \end{bmatrix} \quad (5.40)$$

$$\mathbf{R}_{ki}(:, :, 8) = \begin{bmatrix} 0.9912 & -0.1302 & 0.0222 \\ 0.1307 & 0.9425 & -0.3077 \\ 0.0191 & 0.3079 & 0.9512 \end{bmatrix} \quad (5.41)$$

$$\mathbf{R}_{ki}(:, :, 9) = \begin{bmatrix} 0.9875 & -0.0924 & 0.1277 \\ 0.1454 & 0.8466 & -0.5120 \\ -0.0608 & 0.5242 & 0.8494 \end{bmatrix} \quad (5.42)$$

$$\mathbf{R}_{ki}(:, :, 10) = \begin{bmatrix} 0.9839 & -0.1242 & 0.1285 \\ 0.1769 & 0.7796 & -0.6008 \\ -0.0256 & 0.6138 & 0.7890 \end{bmatrix} \quad (5.43)$$

$$\mathbf{t}_{ki} = \begin{bmatrix} -1443 & 2915 & 8027 \\ -1718 & 3688 & 10441 \\ -1558 & 3208 & 8818 \\ -1608 & 3456 & 9558 \\ -1576 & 3621 & 10081 \\ -1636 & 3863 & 10698 \\ -1872 & 3802 & 10449 \\ -1450 & 3859 & 10753 \\ -1594 & 4184 & 11528 \\ -1563 & 3799 & 10332 \end{bmatrix} \quad (5.44)$$

\mathbf{R}_{ki} adalah matriks rotasi pada kamera kiri. $\mathbf{R}_{ki}(:, :, 1)$ adalah matriks rotasi kiri pada citra pola planar pertama dari kamera kiri kemudian $\mathbf{R}_{ki}(:, :, 2)$ adalah matriks rotasi pada pola planar kedua dari kamera kiri, begitu pula selanjutnya. \mathbf{t}_{ki} adalah matriks translasi pada kamera kiri. Koefisien distorsi pada kamera kiri adalah

$$k_1 = -0.0712 \quad (5.45)$$

$$k_2 = -0.2097 \quad (5.46)$$

Parameter intrinsik pada kamera kanan yang dihasilkan adalah,

$$\mathbf{A}_{ka} = \begin{bmatrix} 2496 & 0 & 545 \\ 0 & 2396 & -327 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.47)$$

Parameter ekstrinsiknya pada kamera kanan adalah

$$\mathbf{R}_{ka}(:, :, 1) = \begin{bmatrix} 0.9885 & -0.1466 & -0.0368 \\ 0.1125 & 0.8764 & -0.4683 \\ 0.1009 & 0.4588 & 0.8828 \end{bmatrix} \quad (5.48)$$

$$\mathbf{R}_{ka}(:, :, 2) = \begin{bmatrix} 0.8521 & 0.1265 & 0.5079 \\ 0.0273 & 0.9583 & -0.2844 \\ -0.5227 & 0.2562 & 0.8131 \end{bmatrix} \quad (5.49)$$

$$\mathbf{R}_{ka}(:, :, 3) = \begin{bmatrix} 0.9896 & -0.1398 & 0.0337 \\ 0.1365 & 0.8390 & -0.5267 \\ 0.0453 & 0.5258 & 0.8494 \end{bmatrix} \quad (5.50)$$

$$\mathbf{R}_{ka}(:, :, 4) = \begin{bmatrix} 0.9864 & -0.1643 & 0.0024 \\ 0.1610 & 0.9633 & -0.2148 \\ 0.0330 & 0.2123 & 0.9766 \end{bmatrix} \quad (5.51)$$

$$\mathbf{R}_{ka}(:, :, 5) = \begin{bmatrix} 0.9861 & -0.0522 & 0.1577 \\ 0.0990 & 0.9470 & -0.3057 \\ -0.1334 & 0.3170 & 0.9390 \end{bmatrix} \quad (5.52)$$

$$\mathbf{R}_{ka}(:, :, 6) = \begin{bmatrix} 0.9887 & -0.0799 & 0.1265 \\ 0.1181 & 0.9357 & -0.3324 \\ -0.0918 & 0.3437 & 0.9346 \end{bmatrix} \quad (5.53)$$

$$\mathbf{R}_{ka}(:, :, 7) = \begin{bmatrix} 0.9845 & -0.1045 & 0.1407 \\ 0.1410 & 0.9491 & -0.2815 \\ -0.1041 & 0.2970 & 0.9492 \end{bmatrix} \quad (5.54)$$

$$\mathbf{R}_{ka}(:, :, 8) = \begin{bmatrix} 0.9916 & -0.1279 & -0.0183 \\ 0.1242 & 0.9829 & -0.1362 \\ 0.0354 & 0.1328 & 0.9905 \end{bmatrix} \quad (5.55)$$

$$\mathbf{R}_{ka}(:, :, 9) = \begin{bmatrix} 0.9897 & -0.1096 & 0.0923 \\ 0.1355 & 0.9251 & -0.3547 \\ -0.0465 & 0.3635 & 0.9304 \end{bmatrix} \quad (5.56)$$

$$\mathbf{R}_{ka}(:, :, 10) = \begin{bmatrix} 0.9861 & -0.1410 & 0.0874 \\ 0.1654 & 0.8752 & -0.4545 \\ -0.0125 & 0.4627 & 0.8864 \end{bmatrix} \quad (5.57)$$

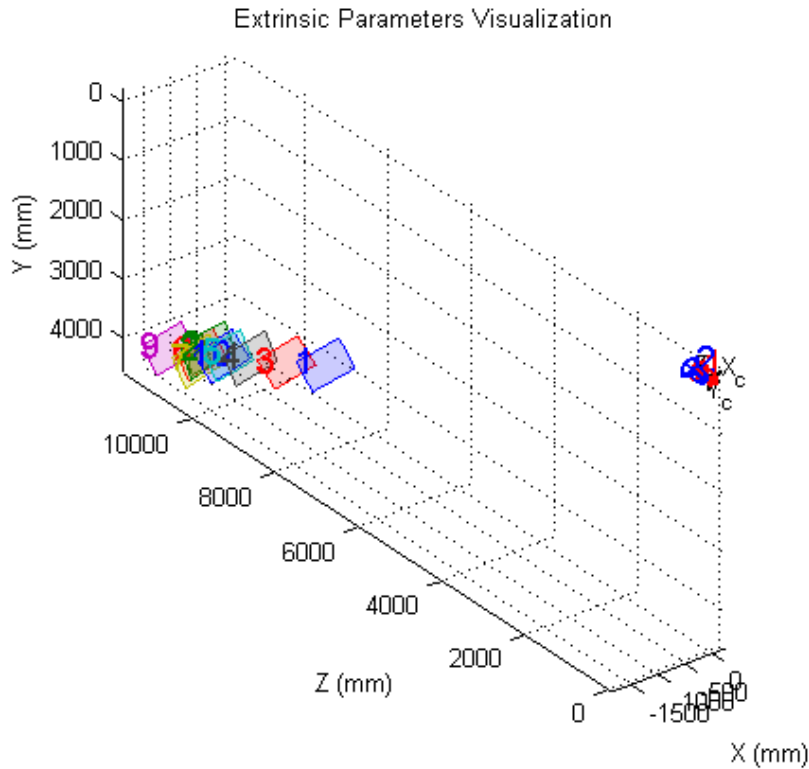
$$\mathbf{t}_{ka} = \begin{bmatrix} -1431 & 1380 & 8508 \\ -1665 & 1708 & 11026 \\ -1533 & 1526 & 9341 \\ -1570 & 1638 & 10114 \\ -1529 & 1706 & 10657 \\ -1579 & 1834 & 11309 \\ -1819 & 1817 & 11056 \\ -1391 & 1821 & 11359 \\ -1522 & 2002 & 12182 \\ -1512 & 1836 & 10936 \end{bmatrix} \quad (5.58)$$

\mathbf{R}_{ka} adalah matriks rotasi pada kamera kanan. \mathbf{t}_{ka} adalah matriks translasi pada kamera kiri. Koefisien distorsi pada kamera kanan adalah

$$k_1 = -0.1219 \quad (5.59)$$

$$k_2 = 0.0568 \quad (5.60)$$

Parameter ekstrinsik yang dihasilkan, dapat divisualisasikan sebagai pada Gambar 5.44 berikut ini,



Gambar 5.44: Visualisasi parameter ekstrinsik pada pola planar di data kedua dengan posisi kamera sejajar dengan obyek

Data yang diambil dibagi menjadi dua interval yaitu jarak 750-1050 cm dan jarak 1050-1350 cm. Hal ini dilakukan karena apabila jaraknya tidak dibagi pasti error yang didapat akan sangat besar. Data dihitung pada interval masing-masing hingga mendapatkan rasio error, pada akhirnya data akan dihitung pada semua keseluruhan data dengan menggunakan rata-rata faktor pada semua jarak yang dihitung. Citra yang diambil kemudian dihilangkan distorsinya agar proyeksi bujur sangkarnya menjadi lurus, kemudian dideteksi koordinat pojok kiri obyek yang akan diestimasi jaraknya. Seperti halnya pada data pertama, uji coba dilakukan empat kali. Data koordinat pojok kiri pada interval pertama di uji coba pertama ditampilkan pada Tabel 5.5 berikut ini,

Tabel 5.5: Jarak dan koordinat pojok kiri obyek citra kiri maupun kanan pada interval jarak pertama di uji coba pertama dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Koordinat citra kiri	Koordinat citra kanan
750	(325,67)	(256,11)
770	(327,66)	(259,12)
790	(328,74)	(260,17)
810	(331,74)	(263,18)

Jarak (cm)	Koordinat citra kiri	Koordinat citra kanan
830	(325,78)	(257,22)
850	(330,79)	(263,23)
870	(323,83)	(256,26)
890	(326,83)	(259,27)
910	(330,88)	(264,33)
930	(327,90)	(261,35)
950	(331,91)	(265,35)
970	(330,94)	(265,38)
990	(333,96)	(268,40)
1010	(335,98)	(270,41)
1030	(335,98)	(271,41)
1050	(338,100)	(275,43)

Data koordinat pojok kiri pada interval kedua di uji coba pertama ditampilkan pada Tabel 5.6 berikut ini,

Tabel 5.6: Jarak dan koordinat pojok kiri obyek citra kiri maupun kanan pada interval jarak kedua di uji coba pertama dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Koordinat citra kiri	Koordinat citra kanan
1050	(338,100)	(275,43)
1070	(332,97)	(269,40)
1090	(335,98)	(272,41)
1110	(339,99)	(276,43)
1130	(338,100)	(275,45)
1150	(338,101)	(274,46)
1170	(339,104)	(277,49)
1190	(339,105)	(278,50)
1210	(335,107)	(274,51)
1230	(342,108)	(280,51)
1250	(343,108)	(283,52)
1270	(337,109)	(277,53)
1290	(338,111)	(278,55)
1310	(339,112)	(278,55)
1330	(340,114)	(280,58)
1350	(342,113)	(282,57)

Menggunakan koordinat yang didapat pada Tabel 5.5 dan Tabel 5.5, maka dapat dihitung dispariti, yaitu dengan mengurangi koordinat x pada kedua citra dan dispariti yang dihasilkan dikalikan dengan jarak sesungguhnya. Hasil perkalian dispariti dan jarak sesungguhnya itulah yang disebut dengan faktor. Faktor disini dihitung tiap jarak terelebih dahulu, kemudian baru dihitung rata-rata faktor pada satu interval. Tabel 5.7 ditunjukkan hasil dispariti dan faktor citra pada interval 750 sampai 1050 cm di uji coba pertama,

Tabel 5.7: Faktor citra pada interval jarak pertama di uji coba pertama dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Dispariti (piksel)	Faktor
750	69	51750
770	68	52360
790	68	53720
810	68	55080
830	68	56440
850	67	56950
870	67	58290
890	67	59630
910	66	60060
930	66	61380
950	66	62700
970	65	63050
990	65	64350
1010	65	65650
1030	64	65920
1050	63	66150

Tabel 5.8 ditunjukkan hasil dispariti dan faktor citra pada interval kedua di uji coba pertama,

Tabel 5.8: Faktor citra pada interval jarak kedua dengan posisi kamera sejajar dengan obyek

Jarak (cm)	Dispariti (piksel)	Faktor
1050	63	66150
1070	63	67410
1090	63	68670
1110	63	69930
1130	63	71190
1150	64	73600
1170	62	72540
1190	61	72590
1210	61	73810
1230	62	76260
1250	60	75000
1270	60	76200
1290	60	77400
1310	61	79910
1330	60	79800
1350	60	81000

Faktor-faktor yang sudah didapatkan pada Tabel 5.7 dan Tabel 5.8 dirata-rata, sehingga didapatkan rata-rata faktor interval jarak pertama pada uji coba pertama adalah 59592.5, uji coba kedua adalah 59420.63, uji coba ketiga adalah 59515.63 dan uji coba keempat adalah 59410.63. Rata-rata faktor interval jarak kedua pada uji coba pertama adalah 73841.25, uji coba kedua adalah 73437.5, uji coba ketiga adalah 73731.25 dan yang keempat adalah 74220.63. Apabila semua dispariti dari kedua interval di uji coba pertama dirata-rata adalah 66735.16129, pada uji coba kedua adalah 66492.9, pada uji coba ketiga adalah 6604.84 dan yang keempat adalah 66837.1. Langkah selanjutnya adalah menghitung jarak estimasi, kemudian dibandingkan dengan jarak yang sesungguhnya untuk mendapatkan error estimasi. Tabel 5.9 berikut ini, ditampilkan hasil perbandingan jarak sesungguhnya dengan jarak hasil estimasi pada interval jarak yang pertama di keempat uji coba,

Tabel 5.9: Estimasi jarak dan presentase error pada interval jarak pertama dengan posisi kamera sejajar dengan obyek

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
750	1. 863.6594	0.1515
	2. 861.1684	0.1482
	3. 862.5452	0.1500
	4. 848.7232	0.1316
770	1. 876.3602	0.1381
	2. 861.1684	0.1184
	3. 875.2297	0.1366
	4. 861.0235	0.1182
790	1. 876.3602	0.1093
	2. 873.8327	0.1061
	3. 862.5452	0.0918
	4. 886.7257	0.1224
810	1. 876.3602	0.0819
	2. 886.8750	0.0949
	3. 875.2297	0.0805
	4. 900.1609	0.1113
830	1. 876.3602	0.0558
	2. 873.8327	0.0528
	3. 875.2297	0.0544
	4. 873.6856	0.0526
850	1. 889.4402	0.0464
	2. 848.8660	0.0013
	3. 850.2232	0.0002
	4. 873.6856	0.0278

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
870	1. 889.4402 2. 943.1845 3. 888.2929 4. 886.7257	0.0223 0.0841 0.0210 0.0192
890	1. 889.4402 2. 900.3125 3. 915.6250 4. 886.7257	0.0006 0.0115 0.0287 0.0036
910	1. 902.9166 2. 914.1634 3. 915.6250 4. 914.0096	0.0077 0.0045 0.0061 0.0044
930	1. 902.9166 2. 886.8750 3. 901.7518 4. 900.1609	0.0291 0.0463 0.0303 0.0320
950	1. 902.9166 2. 914.1634 3. 915.6250 4. 914.0096	0.0495 0.0377 0.0361 0.0378
970	1. 916.8076 2. 914.1634 3. 929.9316 4. 914.0096	0.0548 0.0575 0.0413 0.0577
990	1. 916.8076 2. 914.1634 3. 915.6250 4. 914.0096	0.07393 0.07660 0.07512 0.07675
1010	1. 916.8076 2. 928.4472 3. 929.9316 4. 928.2910	0.0922 0.0807 0.0792 0.0808
1030	1. 931.1328 2. 928.4472 3. 929.9316 4. 928.2910	0.0959 0.0985 0.0971 0.0987
1050	1. 945.9126 2. 928.4472 3. 929.9316 4. 943.0257	0.0991 0.1157 0.1143 0.1018

Tabel 5.10 berikut ini, ditampilkan hasil perbandingan jarak sesungguhnya dengan jarak hasil estimasi pada interval jarak yang kedua di keempat uji coba,

Tabel 5.10: Estimasi jarak dan presentase error pada interval jarak kedua dengan posisi kamera sejajar dengan obyek

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
1050	1. 1172.0833 2. 1147.4609 3. 1152.0507 4. 1178.1051	0.1162 0.0928 0.0971 0.1220
1070	1. 1172.0833 2. 1266.1637 3. 1249.6822 4. 1159.6972	0.0954 0.1833 0.1679 0.0838
1090	1. 1172.0833 2. 1184.4758 3. 1189.2137 4. 1178.1051	0.0753 0.0866 0.0910 0.0808
1110	1. 1172.0833 2. 1184.4758 3. 1170.3373 4. 1178.1051	0.0559 0.0670 0.0543 0.0613
1130	1. 1172.0833 2. 1165.6746 3. 1170.3373 4. 1178.1051	0.03724 0.03157 0.03569 0.04257
1150	1. 1153.7695 2. 1165.6746 3. 1170.3373 4. 1178.1051	0.0032 0.0136 0.0176 0.0244
1170	1. 1190.9879 2. 1223.9583 3. 1208.7090 4. 1178.1051	0.0179 0.0461 0.0330 0.0069
1190	1. 1210.5122 2. 1184.4758 3. 1189.2137 4. 1197.1068	0.0172 0.0046 0.0006 0.0059
1210	1. 1210.5122 2. 1184.4758 3. 1189.2137 4. 1197.1068	0.0004 0.0210 0.0171 0.0106
1230	1. 1190.9879 2. 1184.4758 3. 1189.2137 4. 1216.7315	0.0317 0.0370 0.0331 0.0107

Jarak Sesungguhnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
1250	1. 1230.6875 2. 1203.8934 3. 1228.8541 4. 1216.7315	0.0154 0.0368 0.0169 0.0266
1270	1. 1230.6875 2. 1203.8934 3. 1189.2137 4. 1216.7315	0.0309 0.0520 0.0636 0.0419
1290	1. 1230.6875 2. 1203.8934 3. 1208.7090 4. 1216.7315	0.0459 0.0667 0.0630 0.0567
1310	1. 1210.5122 2. 1203.8934 3. 1208.7090 4. 1216.7315	0.0759 0.0809 0.0773 0.0711
1330	1. 1230.6875 2. 1244.7033 3. 1228.8541 4. 1237.0104	0.0746 0.0641 0.0760 0.0699
1350	1. 1230.6875 2. 1244.7033 3. 1249.6822 4. 1237.0104	0.0883 0.0779 0.0743 0.0836

Merujuk pada Tabel 5.9 dan Tabel 5.10 median rasio error relatif yang dihasilkan pada interval 750 cm sampai 1059 cm pada uji coba pertama adalah 0.0648, uji coba kedua adalah 0.0786, uji coba ketiga adalah 0.06481 dan yang keempat adalah 0.06724. Interval jarak 1050 sampai 1350 didapatkan median dari rasio error relatif pada uji coba pertama adalah 0.0416, uji coba kedua adalah 0.058, uji coba ketiga adalah 0.0586 dan yang keempat adalah 0.0496. Faktor semua dispariti digunakan pada semua data, sehingga didapatkan median rasio error relatif pada uji coba pertama adalah 0.1101, pada uji coba kedua adalah 0.1136, pada uji coba ketiga adalah 0.1121 dan yang keempat adalah 0.1091. Dari median rasio error relatif yang didapat pada interval jarak ini, dapat disimpulkan bahwa sistem yang digunakan cukup stabil dalam mengestimasi jarak.

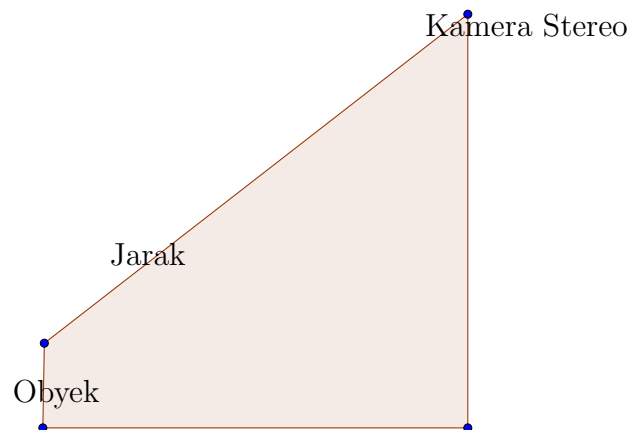
Perancangan GMM pada interval jarak ini menggunakan inisialisasi nilai varian adalah 0.00009, sedangkan laju pelatihan adalah 0.001, kemudian banyak model *background* yang digunakan adalah tujuh. Morfologi citra menggunakan filtering, erosi dan filling. Berikut ini langsung ditunjukkan hasil simulasi deteksi jarak obyek bergerak di interval jarak kedua pada Gambar 5.45 berikut ini,



Gambar 5.45: Deteksi obyek bergerak beserta estimasi jarak dari kamera ke obyek dengan posisi kamera sejajar dengan obyek

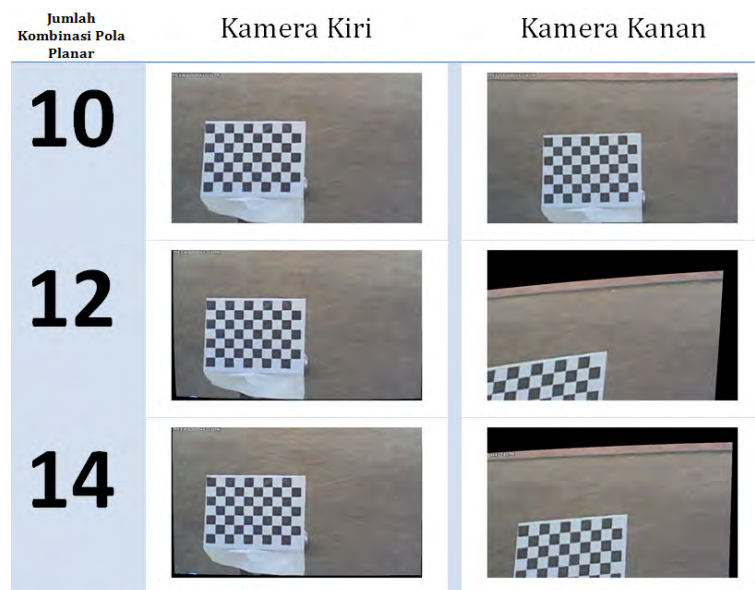
5.3.2 Uji Coba Kamera di atas Obyek

Pengambilan data pada posisi kamera di atas obyek dilakukan di samping tribun, pengambilan data dilakukan dengan menaruh kamera pada tribun mengarah ke bawah pada obyek. Visualisasi pengukuran jarak dari obyek ditunjukkan pada Gambar 5.46 berikut ini,



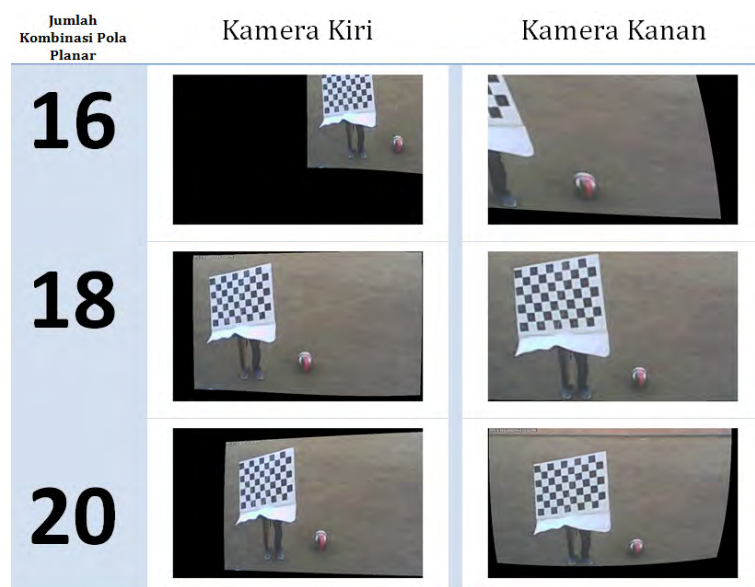
Gambar 5.46: Visualiasi Pengambilan data jarak sesungguhnya di atas tribun

Kombinasi citra yang digunakan untuk posisi menghadap kamera berjumlah 24 citra, dengan rincian 12 citra dari kamera kiri dan 12 dari kamera kanan. Hasil uji coba kombinasi ini sangat bagus terutama ketika kombinasi jumlah lebih dari 14 citra. Berikut ini ditunjukkan tiga hasil di bawah kombinasi jumlah 14 citra pada Gambar 5.47,



Gambar 5.47: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar sepuluh sampai 14 dengan posisi menghadap kamera

Kombinasi citra dengan posisi miring dari kamera berjumlah 24 citra, yaitu 12 citra dari kamera kiri dan 12 dari kanan. Uji coba pada kombinasi ini didapatkan hasil yang kurang bagus. Tiga kombinasi uji coba ditunjukkan pada Gambar 5.48 berikut ini,



Gambar 5.48: Hasil penghilangan distorsi dengan jumlah kombinasi pola planar 16 sampai 20 dengan posisi miring dari kamera

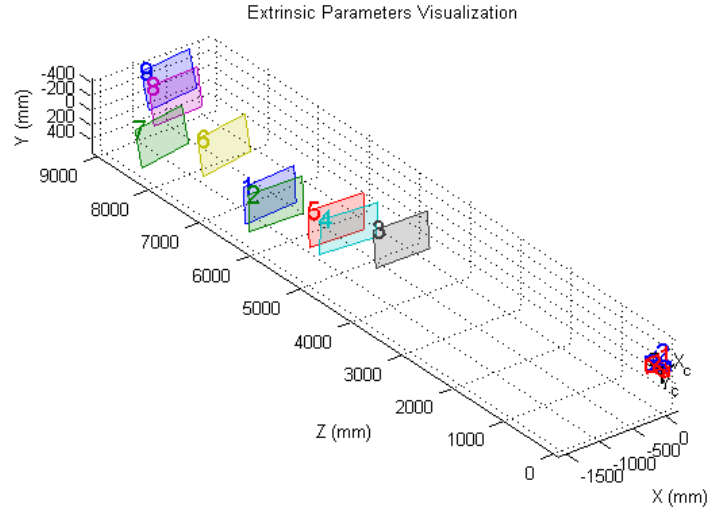
Kombinasi gabungan citra pola planar posisi miring dan menghadap

kamera ternyata didapatkan hasil yang bagus juga seperti halnya pada saat kombinasi semua citra menghadap kamera. Hasil terbaik pada kombinasi gabungan ini didapatkan ketika kombinasi jumlah melebihi 8 citra pola planar. Sekian banyak hasil bagus pada kombinasi pola planar di atas tribun dipilih kombinasi pola planar gabungan dengan total jumlah 18 citra untuk estimasi jarak. Koefisien distorsi yang dihasilkan pada kombinasi pola planar yang dipilih, untuk kamera kiri adalah $k_1 = -0.2623$ dan $k_2 = 7.4080$. Koefisien distorsi pada kamera kanan adalah $k_1 = -0.0175$ dan $k_2 = -0.6443$. Parameter intrinsik pada kamera kiri dan kanan adalah,

$$\mathbf{A}_{ki} = \begin{bmatrix} 2324 & 0 & 627 \\ 0 & 2202 & -676 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_{ka} = \begin{bmatrix} 2496 & 0 & 545 \\ 0 & 2396 & -327 \\ 0 & 0 & 1 \end{bmatrix}$$

Parameter ekstrinsik yang dihasilkan, dapat divisualisasikan sebagai pada Gambar 5.49,



Gambar 5.49: Visualisasi parameter ekstrinsik pada pola planar di samping tribun

Data yang diambil dimulai dari jarak 480 cm sampai 700 cm dengan interval 20 cm. Seperti pada data posisi kamera sejajar dengan obyek, uji coba dilakukan empat kali untuk melihat kestabilan sistem. Citra yang diambil kemudian dihilangkan distorsinya kemudian dideteksi koordinat pojok kiri obyek yang akan diestimasi jaraknya, data koordinat pojok kiri pada uji coba pertama ditampilkan pada Tabel 5.11 berikut ini,

Tabel 5.11: Jarak dan koordinat pojok kiri obyek citra kiri maupun kanan pada uji coba pertama dengan posisi kamera di atas obyek

Jarak (cm)	Koordinat citra kiri	Koordinat citra kanan
480	(209,220)	(282,266)
500	(201,198)	(272,247)
520	(25,168)	(332,213)
540	(209,139)	(280,186)
560	(198,120)	(267,170)
580	(250,97)	(321,142)
600	(192,82)	(262,132)
620	(222,56)	(290,100)
640	(203,40)	(272,88)
660	(234,24)	(302,71)
680	(222,13)	(290,62)
700	(1,232)	(300,43)

Tabel 5.12 ditampilkan hasil dispariti dan faktor citra pada uji coba pertama,

Tabel 5.12: Faktor citra pada uji coba pertama dengan posisi kamera di atas obyek

Jarak (cm)	Dispariti (piksel)	Faktor
480	73	35040
500	71	35500
520	73	37960
540	71	38340
560	69	38640
580	71	41180
600	70	42000
620	68	42160
640	69	44160
660	68	44880
680	68	46240
700	68	47600

Faktor-faktor yang sudah didapatkan pada Tabel 5.12 dirata-rata, sehingga didapatkan rata-rata faktor interval pada uji coba pertama adalah 41141.67, pada uji coba kedua adalah 41936.67, pada uji coba ketiga adalah 41847.67, pada uji keempat adalah 41818.33. Dengan menggunakan persamaan (4.22), estimasi jarak bisa didapatkan. Jarak estimasi dan jarak yang sesungguhnya pada data di atas tribun dibandingkan, dan hasilnya ditampilkan pada Tabel 5.13 berikut ini,

Tabel 5.13: Estimasi jarak dan presentase error dengan posisi kamera di atas obyek

Jarak Sebenarnya (cm)	Jarak Estimasi (cm) ke-	Rasio Error Relatif
480	1. 563.5844 2. 574.4748 3. 581.1342 4. 580.8101	0.1741 0.1968 0.2106 0.2100
500	1. 579.4600 2. 590.6572 3. 581.1342 4. 580.8101	0.1589 0.1813 0.1622 0.1616
520	1. 563.5844 2. 566.7117 3. 573.1735 4. 565.1126	0.0838 0.0898 0.1022 0.0867
540	1. 579.4600 2. 582.4537 3. 606.4009 4. 565.1126	0.0730 0.0786 0.1229 0.0465
560	1. 596.2560 2. 582.4537 3. 581.1342 4. 597.4047	0.0647 0.0400 0.0377 0.0667
580	1. 579.4600 2. 574.4748 3. 589.3192 4. 606.0628	0.0009 0.0095 0.0160 0.0449
600	1. 587.7380 2. 582.4537 3. 581.1342 4. 588.9906	0.0204 0.0292 0.0314 0.0183
620	1. 605.0245 2. 582.4537 3. 589.3192 4. 597.4047	0.0241 0.0605 0.0494 0.0364
640	1. 596.2560 2. 599.0952 3. 597.7380 4. 597.4047	0.0683 0.0639 0.0660 0.0665
660	1. 605.0245 2. 607.7777 3. 597.7380 4. 597.4047	0.0832 0.0791 0.0943 0.0948

BAB 6

KESIMPULAN DAN SARAN

Dari analisis dan pembahasan yang sudah dilakukan, dapat ditarik kesimpulan serta saran untuk pengembangan dan perbaikan penelitian selanjutnya.

6.1 Kesimpulan

Penelitian ini telah berhasil mengkonfigurasi dua web cam menjadi kamera stereo dan telah berhasil melakukan kalibrasi kamera dengan beberapa kombinasi jumlah maupun posisi pola planar. Hasil yang bagus dalam melakukan kalibrasi kamera pada posisi kamera sejajar adalah pada kombinasi posisi pola planar yang digunakan adalah menghadap kamera. Hasil yang bagus dalam melakukan kalibrasi kamera pada posisi kamera di atas tribun, adalah pada kombinasi menghadap kamera dengan hasil terbaik pada kombinasi jumlah lebih dari 14 pola planar. Kombinasi pola panar gabungan posisi miring dan menghadap kamera pada posisi kamera di atas tribun juga menghasilkan kalibrasi kamera yang bagus, terutama ketika kombinasi jumlah lebih dari delapan citra.

Tiga data yang telah digunakan untuk melakukan uji coba, didapat kesimpulan bahwa estimasi jarak terbaik adalah ketika kamera sejajar dengan obyek pada interval jarak 750 sampai 1050 cm dengan median rasio error relatif 0.0648 dan interval jarak 1050 sampai 1350 cm dengan median rasio error relatif 0.0496. Estimasi jarak terbaik tersebut menggunakan parameter intrinsik dan ekstrinsik pada persamaan (5.33-5.60) dengan rata-rata faktor 59592.5 untuk interval jarak 750 sampai 1050 cm dan 74220.63 untuk jarak 1050 sampai 1350 cm.

6.2 Saran

Beberapa hal yang bisa dikembangkan lebih jauh, diantaranya adalah estimasi jarak dilakukan pada lebih dari satu obyek yang bergerak. Deteksi obyek bergerak pada posisi kamera sejajar dengan obyek juga perlu diperbaiki, sehingga estimasi jarak obyek bergerak bisa lebih optimal. Tambahan yang dapat menyempurnakan tesis ini adalah dapat dikembangkannya peta dispariti, yang mewakili dispariti pada kedua citra. Pengembangan peta dispariti, dapat membuat perhitungan rata-rata faktor lebih akurat bila dilakukan secara otomatis dalam program, dan dalam kondisi estimasi jarak ketika obyek bergerak juga lebih efisien, karena hanya perlu mendeteksi obyek pada salah satu video tidak pada kedua-duanya.

LAMPIRAN A

Contoh Perhitungan Kalibrasi Kamera Zhang

Deteksi setiap titik pola planar seperti ditunjukkan pada Gambar 5.3 dan Gambar 5.4. Berikut ini koordinat setiap titik pada pola planar kiri maupun kanan, yang ditampilkan pada berikut Tabel 1 Lampiran ini,

Tabel 1 Lampiran. Koordinat Titik Pola Planar Kiri dan Kanan

No Titik	Koordinat Titik Pola Kiri	Koordinat Titik Pola Kanan
Titik 1	(198.2897, 202.9816)	(142.3913, 276.2516)
Titik 2	(196.9440, 241.0734)	(140.1024, 313.1302)
Titik 3	(195.3491, 279.7053)	(137.7794, 350.3179)
Titik 4	(193.9997, 318.3077)	(135.2941, 388.1032)
Titik 5	(192.0159, 357.2714)	(132.6543, 426.3098)
Titik 6	(190.4427, 396.5674)	(129.9961, 464.9395)
Titik 7	(236.8217, 203.9243)	(179.7625, 277.7024)
Titik 8	(235.2758, 242.1439)	(177.3155, 314.3733)
Titik 9	(233.9262, 280.7838)	(175.1252, 351.7441)
Titik 10	(232.7939, 319.6512)	(172.8729, 389.4344)
Titik 11	(231.1071, 358.3291)	(170.6435, 427.7553)
Titik 12	(229.3608, 397.7735)	(167.9326, 466.2278)
Titik 13	(275.0677, 205.0293)	(216.7501, 278.9235)
Titik 14	(273.9736, 243.1159)	(214.7512, 315.7438)
Titik 15	(272.6947, 281.7531)	(212.6174, 352.8927)
Titik 16	(271.1872, 320.7017)	(210.6744, 390.7542)
Titik 17	(269.9085, 359.7126)	(208.2785, 428.9033)
Titik 18	(268.2687, 398.9325)	(206.0387, 467.6855)
Titik 19	(313.8037, 206.0962)	(253.9291, 280.1624)
Titik 20	(312.3815, 244.2880)	(252.0227, 316.9093)
Titik 21	(311.2015, 282.8410)	(250.0977, 354.0063)
Titik 22	(310.0815, 321.7681)	(248.3371, 391.9247)
Titik 23	(308.9568, 360.7690)	(246.7031, 429.9663)
Titik 24	(307.7126, 400.0329)	(244.6168, 468.9315)
Titik 25	(352.0641, 207.1237)	(290.7692, 281.3502)
Titik 26	(351.0845, 245.1940)	(289.1199, 317.9870)
Titik 27	(350.0180, 283.8783)	(287.7222, 355.1493)
Titik 28	(348.9651, 322.9082)	(285.9918, 393.1374)
Titik 29	(347.9023, 361.9368)	(284.2994, 431.1614)
Titik 30	(346.7863, 401.1623)	(282.7303, 470.0835)

No Titik	Koordinat Titik Pola Kiri	Koordinat Titik Pola Kanan
Titik 31	(390.7118, 208.0856)	(327.7353, 282.7281)
Titik 32	(389.7183, 246.3086)	(326.3250, 319.1214)
Titik 33	(388.7300, 284.8502)	(325.0187, 356.2583)
Titik 34	(387.8012, 324.0053)	(323.6727, 394.3525)
Titik 35	(386.8793, 363.0496)	(322.1415, 432.6305)
Titik 36	(385.7174, 402.4781)	(320.8712, 471.3275)
Titik 37	(429.2032, 208.9154)	(364.8836, 283.7597)
Titik 38	(428.6964, 247.2938)	(363.8109, 320.2028)
Titik 39	(427.8036, 285.9945)	(362.7648, 357.7184)
Titik 40	(426.9673, 325.0684)	(361.5848, 395.7324)
Titik 41	(426.0879, 364.2098)	(360.2373, 434.0127)
Titik 42	(425.1401, 403.9160)	(359.1593, 472.9031)
Titik 43	(468.0283, 209.8679)	(401.7868, 284.7817)
Titik 44	(467.2866, 248.1924)	(400.9962, 321.6580)
Titik 45	(466.7681, 287.0087)	(400.0258, 358.8463)
Titik 46	(465.9545, 326.1721)	(399.0923, 396.9378)
Titik 47	(465.3135, 365.4635)	(398.2409, 435.1186)
Titik 48	(464.7443, 405.1408)	(397.3509, 474.2029)
Titik 49	(506.8533, 210.8384)	(438.8025, 285.9471)
Titik 50	(506.3415, 249.2160)	(438.1210, 322.7333)
Titik 51	(505.8939, 288.0714)	(437.6999, 359.9727)
Titik 52	(505.2487, 327.3213)	(437.0393, 398.0932)
Titik 53	(504.7968, 366.8463)	(436.2847, 436.6370)
Titik 54	(504.1715, 406.4149)	(435.8807, 475.7003)

Setelah didapatkan titik citra, langkah selanjutnya adalah membuat titik dunia, yaitu titik representasi sesungguhnya, dengan titik sudut paling kiri memiliki koordinat (0,0) . Karena jarak antar satu titik pola planar ke titik disampingnya adalah 100 mm, sehingga setiap menambah satu titik dunia kesamping maupun ke bawah, ditambah 100. Dimisalkan **titikDunia** sebagai titik dunia, berikut titik dunianya jika disusun dalam matriks,

$$\begin{bmatrix} (0, 0) & (0, 100) & (0, 200) & (0, 300) & (0, 400) & (0, 500) \\ (100, 0) & (100, 100) & (100, 200) & (100, 300) & (100, 400) & (100, 500) \\ (200, 0) & (200, 100) & (200, 200) & (200, 300) & (200, 400) & (200, 500) \\ (300, 0) & (300, 100) & (300, 200) & (300, 300) & (300, 400) & (300, 500) \\ (400, 0) & (400, 100) & (400, 200) & (400, 300) & (400, 400) & (400, 500) \\ (500, 0) & (500, 100) & (500, 200) & (500, 300) & (500, 400) & (500, 500) \\ (600, 0) & (600, 100) & (600, 200) & (600, 300) & (600, 400) & (600, 500) \\ (700, 0) & (700, 100) & (700, 200) & (700, 300) & (700, 400) & (700, 500) \\ (800, 0) & (800, 100) & (800, 200) & (800, 300) & (800, 400) & (800, 500) \end{bmatrix}^T$$

Setelah titik citra dan titik dunia didapatkan, maka dapat disusun matriks **X** seperti pada persamaan (4.1) sebagai langkah awal untuk mendapatkan homografi. Sebagai langkah awal, titik citra yang digunakan adalah citra pola

kiri sehingga matriks homografi yang didapatkan adalah matriks homografi untuk pola planar kiri. Berikut matriks \mathbf{X} ,

$$\mathbf{X} = \begin{bmatrix} 504 & 406 & 1 & 0 & 0 & 0 & 0 & 0 \\ 505 & 367 & 1 & 0 & 0 & 0 & 0 & 0 \\ 505 & 327 & 1 & 0 & 0 & 0 & 0 & 0 \\ 506 & 288 & 1 & 0 & 0 & 0 & 0 & 0 \\ 506 & 249 & 1 & 0 & 0 & 0 & 0 & 0 \\ 507 & 211 & 1 & 0 & 0 & 0 & 0 & 0 \\ 465 & 405 & 1 & 0 & 0 & 0 & -46474 & -40514 \\ 465 & 365 & 1 & 0 & 0 & 0 & -46531 & -36546 \\ 466 & 326 & 1 & 0 & 0 & 0 & -46595 & -32617 \\ 467 & 287 & 1 & 0 & 0 & 0 & -46677 & -28701 \\ 467 & 248 & 1 & 0 & 0 & 0 & -46729 & -24819 \\ 468 & 210 & 1 & 0 & 0 & 0 & -46803 & -20987 \\ 425 & 404 & 1 & 0 & 0 & 0 & -85028 & -80783 \\ 426 & 364 & 1 & 0 & 0 & 0 & -85218 & -72842 \\ 427 & 325 & 1 & 0 & 0 & 0 & -85393 & -65014 \\ 428 & 286 & 1 & 0 & 0 & 0 & -85561 & -57199 \\ 429 & 247 & 1 & 0 & 0 & 0 & -85739 & -49459 \\ 429 & 209 & 1 & 0 & 0 & 0 & -85841 & -41783 \\ 386 & 402 & 1 & 0 & 0 & 0 & -115715 & -120743 \\ 387 & 363 & 1 & 0 & 0 & 0 & -116064 & -108915 \\ 388 & 324 & 1 & 0 & 0 & 0 & -116340 & -97202 \\ 389 & 285 & 1 & 0 & 0 & 0 & -116619 & -85455 \\ 390 & 246 & 1 & 0 & 0 & 0 & -116915 & -73893 \\ 391 & 208 & 1 & 0 & 0 & 0 & -117214 & -62426 \\ 347 & 401 & 1 & 0 & 0 & 0 & -138715 & -160465 \\ 348 & 362 & 1 & 0 & 0 & 0 & -139161 & -144775 \\ 349 & 323 & 1 & 0 & 0 & 0 & -139586 & -129163 \\ 350 & 284 & 1 & 0 & 0 & 0 & -140007 & -113551 \\ 351 & 245 & 1 & 0 & 0 & 0 & -140434 & -98078 \\ 352 & 207 & 1 & 0 & 0 & 0 & -140826 & -82849 \\ 308 & 400 & 1 & 0 & 0 & 0 & -153856 & -200016 \\ 309 & 361 & 1 & 0 & 0 & 0 & -154478 & -180384 \\ 310 & 322 & 1 & 0 & 0 & 0 & -155041 & -160884 \\ 311 & 283 & 1 & 0 & 0 & 0 & -155601 & -141420 \\ 312 & 244 & 1 & 0 & 0 & 0 & -156191 & -122144 \\ 314 & 206 & 1 & 0 & 0 & 0 & -156902 & -103048 \\ 268 & 399 & 1 & 0 & 0 & 0 & -160961 & -239359 \\ 270 & 360 & 1 & 0 & 0 & 0 & -161945 & -215828 \\ 271 & 321 & 1 & 0 & 0 & 0 & -162712 & -192421 \\ 273 & 282 & 1 & 0 & 0 & 0 & -163617 & -169052 \\ 274 & 243 & 1 & 0 & 0 & 0 & -164384 & -145870 \end{bmatrix}$$

275	205	1	0	0	0	-165041	-123018
229	398	1	0	0	0	-160553	-278441
231	358	1	0	0	0	-161775	-250830
233	320	1	0	0	0	-162956	-223756
234	281	1	0	0	0	-163748	-196549
235	242	1	0	0	0	-164693	-169501
237	204	1	0	0	0	-165775	-142747
190	397	1	0	0	0	-152354	-317254
192	357	1	0	0	0	-153613	-285817
194	318	1	0	0	0	-155200	-254646
195	280	1	0	0	0	-156279	-223764
197	241	1	0	0	0	-157555	-192859
198	203	1	0	0	0	-158632	-162385
0	0	0	504	406	1	0	0
0	0	0	505	367	1	-50480	-36685
0	0	0	505	327	1	-101050	-65464
0	0	0	506	288	1	-151768	-86421
0	0	0	506	249	1	-202537	-99686
0	0	0	507	211	1	-253427	-105419
0	0	0	465	405	1	0	0
0	0	0	465	365	1	-46531	-36546
0	0	0	466	326	1	-93191	-65234
0	0	0	467	287	1	-140030	-86103
0	0	0	467	248	1	-186915	-99277
0	0	0	468	210	1	-234014	-104934
0	0	0	425	404	1	0	0
0	0	0	426	364	1	-42609	-36421
0	0	0	427	325	1	-85393	-65014
0	0	0	428	286	1	-128341	-85798
0	0	0	429	247	1	-171479	-98918
0	0	0	429	209	1	-214602	-104458
0	0	0	386	402	1	0	0
0	0	0	387	363	1	-38688	-36305
0	0	0	388	324	1	-77560	-64801
0	0	0	389	285	1	-116619	-85455
0	0	0	390	246	1	-155887	-98523
0	0	0	391	208	1	-195356	-104043
0	0	0	347	401	1	0	0
0	0	0	348	362	1	-34790	-36194
0	0	0	349	323	1	-69793	-64582
0	0	0	350	284	1	-105005	-85163
0	0	0	351	245	1	-140434	-98078
0	0	0	352	207	1	-176032	-103562
0	0	0	308	400	1	0	0

$$\begin{bmatrix} 0 & 0 & 0 & 309 & 361 & 1 & -30896 & -36077 \\ 0 & 0 & 0 & 310 & 322 & 1 & -62016 & -64354 \\ 0 & 0 & 0 & 311 & 283 & 1 & -93360 & -84852 \\ 0 & 0 & 0 & 312 & 244 & 1 & -124953 & -97715 \\ 0 & 0 & 0 & 314 & 206 & 1 & -156902 & -103048 \\ 0 & 0 & 0 & 268 & 399 & 1 & 0 & 0 \\ 0 & 0 & 0 & 270 & 360 & 1 & -26991 & -35971 \\ 0 & 0 & 0 & 271 & 321 & 1 & -54237 & -64140 \\ 0 & 0 & 0 & 273 & 282 & 1 & -81808 & -84526 \\ 0 & 0 & 0 & 274 & 243 & 1 & -109589 & -97246 \\ 0 & 0 & 0 & 275 & 205 & 1 & -137534 & -102515 \\ 0 & 0 & 0 & 229 & 398 & 1 & 0 & 0 \\ 0 & 0 & 0 & 231 & 358 & 1 & -23111 & -35833 \\ 0 & 0 & 0 & 233 & 320 & 1 & -46559 & -63930 \\ 0 & 0 & 0 & 234 & 281 & 1 & -70178 & -84235 \\ 0 & 0 & 0 & 235 & 242 & 1 & -94110 & -96858 \\ 0 & 0 & 0 & 237 & 204 & 1 & -118411 & -101962 \\ 0 & 0 & 0 & 190 & 397 & 1 & 0 & 0 \\ 0 & 0 & 0 & 192 & 357 & 1 & -19202 & -35727 \\ 0 & 0 & 0 & 194 & 318 & 1 & -38800 & -63662 \\ 0 & 0 & 0 & 195 & 280 & 1 & -58605 & -83912 \\ 0 & 0 & 0 & 197 & 241 & 1 & -78778 & -96429 \\ 0 & 0 & 0 & 198 & 203 & 1 & -99145 & -101491 \end{bmatrix}$$

Setelah didapatkan matriks \mathbf{X} , kemudian disusun matriks \mathbf{U} seperti pada persamaan (4.2) sebagai berikut

$$\mathbf{U} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 200 \\ 200 \\ 200 \\ 200 \end{bmatrix}$$

200
200
300
300
300
300
300
300
400
400
400
400
400
400
500
500
500
500
500
500
600
600
600
600
600
600
700
700
700
700
700
700
800
800
800
800
800
800
0
100
200
300
400

500
0
100
200
300
400
500
0
100
200
300
400
500
0
100
200
300
400
500
0
100
200
300
400
500
0
100
200
300
400
500
0
100
200
300
400
500

$$\begin{bmatrix} 0 \\ 100 \\ 200 \\ 300 \\ 400 \\ 500 \end{bmatrix}$$

Setelah didapatkan matriks \mathbf{U} dan \mathbf{X} , kemudian dengan persamaan (2.2-4.1) dihitung matriks homografi. Pertama-tama dihitung matriks \mathbf{tvec} seperti pada persamaan (2.2), sehingga didapatkan matriks \mathbf{tvec} ,

$$\mathbf{tvec} = \begin{bmatrix} -2.7 \\ -0 \\ 1354 \\ 0 \\ -2.6 \\ 1032 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Kemudian berikut ini invers dari matriks \mathbf{T} seperti berikut,

$$\mathbf{T} = \begin{bmatrix} \mathbf{tvec}(1,1) & \mathbf{tvec}(4,1) & \mathbf{tvec}(7,1) \\ \mathbf{tvec}(2,1) & \mathbf{tvec}(5,1) & \mathbf{tvec}(8,1) \\ \mathbf{tvec}(3,1) & \mathbf{tvec}(6,1) & \mathbf{tvec}(9,1) \end{bmatrix}^{-1} = \begin{bmatrix} -0.3714 & -0.0075 & 0 \\ 0.0219 & -0.3651 & 0 \\ 480.1461 & 386.8651 & 0.9523 \end{bmatrix}$$

Setiap elemen \mathbf{T} dibagi dengan $\mathbf{T}(3,3)$ yaitu 0.9523, sehingga didapatkan matriks,

$$\begin{bmatrix} -0.3900 & -0.0079 & 0.0000 \\ 0.0230 & -0.3834 & 0.0000 \\ 504.1766 & 406.2271 & 1.0000 \end{bmatrix}$$

Sehingga matriks homografi dari pola planar kiri adalah tranpose dari matriks \mathbf{T} yang setiap elemennya telah dibagi dengan $\mathbf{T}(3,3)$, matriks homografi ditampilkan sebagai berikut,

$$\mathbf{H} = \begin{bmatrix} -0.3900 & 0.0230 & 504.1766 \\ -0.0079 & -0.3834 & 406.2271 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

Dengan cara yang sama, maka didapatkan homografi dari pola planar kanan adalah

$$\mathbf{H} = \begin{bmatrix} -0.3820 & 0.0352 & 435.7357 \\ -0.0118 & -0.3598 & 475.3133 \\ 0.0000 & 0.0001 & 1.0000 \end{bmatrix}$$

Setelah didapatkan matriks \mathbf{H} baik pada pola planar kiri maupun kanan, maka dapat dihitung matriks \mathbf{V} seperti pada persamaan (4.4). Sehingga matriks \mathbf{V} dihasilkan sebagai berikut,

$$\mathbf{V} = \begin{bmatrix} -0.0090 & 0.1493 & 0.0030 & -0 & -0 & 0 \\ 0.1516 & 0.0238 & -0.1469 & -0 & 0 & -0 \\ -0.0134 & 0.1370 & 0.0042 & -0 & -0 & 0 \\ 0.1447 & 0.0343 & -0.1293 & -0 & 0 & -0 \end{bmatrix}$$

Matriks \mathbf{V} didekomposisikan dengan metode SVD, dan dimisalkan \mathbf{D} adalah singular kanan dari \mathbf{V} . Berikut nilai dari \mathbf{D} ,

$$\mathbf{D} = \begin{bmatrix} 0.7139 & -0.1721 & -0.6787 & 0.0007 & -0.0018 & 0 \\ 0.2046 & 0.9783 & -0.0329 & 0.0001 & 0 & 0 \\ -0.6697 & 0.1154 & -0.7336 & 0.0006 & -0.0017 & 0 \\ -0 & -0.0001 & -0.0009 & 0.7557 & 0.6549 & -0.0001 \\ 0.0002 & -0 & -0.0025 & -0.6549 & 0.7557 & 0 \\ -0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Kolom terakhir dari matriks \mathbf{D} disusun seperti pada persamaan (2.7), sehingga didapatkan persamaan \mathbf{B} seperti berikut ini,

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & -0.0001 \\ 0 & 0 & 0 \\ -0 & 0 & 1 \end{bmatrix}$$

Pada fase ini parameter instrinsik dapat dihitung menggunakan persamaan (2.13-2.14), kemudian dihitung parameter ekstrinsik. Parameter-parameter yang dihasilkan, kemudian dioptimalkan agar errornya kecil menggunakan persamaan (4.7), berikut parameter intrinsik yang dihasilkan,

$$\mathbf{A} = \begin{bmatrix} 1010 & 0 & 338 \\ 0 & 1013 & 284 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

Parameter ekstrinsi yang dihasilkan setelah proses optimasi parameter ditampilkan pada \mathbf{R}_{kl} , \mathbf{R}_{kk} dan \mathbf{t} berikut ini,

$$\mathbf{R}_{kl} = \begin{bmatrix} -0.9993 & -0.0273 & 0.0236 \\ 0.0293 & -0.9959 & 0.0855 \\ 0.0212 & 0.0861 & 0.9961 \end{bmatrix} \quad (1.2)$$

$$\mathbf{R}_{kk} = \begin{bmatrix} -0.9994 & -0.0322 & 0.0115 \\ 0.0337 & -0.9854 & 0.1666 \\ 0.0060 & 0.1669 & 0.9860 \end{bmatrix} \quad (1.3)$$

$$\mathbf{t} = \begin{bmatrix} 423 & 311 & 2573 \\ 256 & 499 & 2644 \end{bmatrix} \quad (1.4)$$

$\mathbf{R}_{kl}, \mathbf{R}_{kk}$ adalah matriks rotasi pada kamera kiri dan kanan. Sedangkan \mathbf{t} adalah matriks translasi. Dengan mengoptimalkan parameter didapatkan koefisien ditorsinya sebagai berikut,

$$k_1 = 0.1104 \quad (1.5)$$

$$k_2 = -0.8901 \quad (1.6)$$

LAMPIRAN B

Kode Program Estimasi Jarak pada Obyek Bergerak

Berikut inik dilampirkan kode program estimasi jarak pada obyek bergerak.

```
function varargout = coba_webcamp(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @coba_webcamp_OpeningFcn, ...
                  'gui_OutputFcn',    @coba_webcamp_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
%%

function coba_webcamp_OpeningFcn(hObject, eventdata, handles, varargin)

axes(handles.axes5)
axis off
a = imread('Matematika.png');
imshow(a)
global videoSrc frame videoSrc1 frame1 foreground...
foregroundDetector foreground1 blobAnalysis...
focal_length cm1 cm2 posisi
load data_kalibrasi.mat;
videoSrc = vision.VideoFileReader(uigetfile...
({'*.avi;*.mp4';'*.*'}));
videoSrc1 = vision.VideoFileReader(uigetfile...
({'*.avi;*.mp4';'*.*'}));
frame = videoSrc.step();
frame1 = videoSrc1.step();
cm1 = stereoParams.CameraParameters1;
```

```

cm2 = stereoParams.CameraParameters2;
frame = undistortImage(frame, cm1,'nearest');
frame1 = undistortImage(frame1, cm2,'nearest');
frame = im2uint8(frame);
frame1 = im2uint8(frame1);
for i = 1:150
    frame = step(videoSrc);
    frame1 = step(videoSrc1);
    frame = undistortImage(frame, cm1,'nearest');
    frame1 = undistortImage(frame1, cm2,'nearest');
    foreground = step(foregroundDetector, frame);
    foreground1 = step(foregroundDetector, frame1);
end
axes(handles.axes4);
imshow(cat(3, frame(:,:,1), frame1(:,:,2:3)));
h = imrect;
setColor(h,'red');
posisi = wait(h);
focal_length = f;
% focal_length = str2double(focal_length);
%%
function varargout = coba_webcamp_OutputFcn()
handles.output = hObject;
varargout{1} = handles.output;
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global foreground frame foreground1 frame1 result...
videoSrc foregroundDetector videoSrc1 blobAnalysis...
focal_length cm1 cm2 posisi
pos = round(posisi);
ambl_foreground1 = zeros(size(foreground,1),size(foreground,2));
ambl_foreground2 = zeros(size(foreground1,1),size(foreground1,2));
for i = pos(1):pos(1)+pos(3)
    for j = pos(2):pos(2)+pos(4)
        ambl_foreground1(j,i) = foreground(j,i);
        ambl_foreground2(j,i) = foreground1(j,i);
    end
end
end
deteksi_objek(ambl_foreground1);
deteksi_objek1(ambl_foreground2);
filteredForeground = foreground;
filteredForeground1 = foreground1;
bbox = step(blobAnalysis, filteredForeground);
bbox1 = step(blobAnalysis, filteredForeground1);
if size(bbox,1) ~= 0 && size(bbox1,1) ~= 0

```

```

        jarak = focal_length/abs(bbox(1)-bbox1(1));
        numCars = jarak;
        result = insertObjectAnnotation(frame, 'rectangle', ...
            bbox,numCars,'Color', 'green');
        foreground = uint8(repmat(foreground, [1, 1, 3])) .* 255;
        foreground = insertObjectAnnotation(foreground,...
            'rectangle',bbox,numCars,'Color', 'red');
        axes(handles.axes4);
        imshow(result)
        axes(handles.axes6);
        imshow(foreground)
    end
    clear bbox bbox1
    while ~isDone(videoSrc)
        frame = step(videoSrc);
        frame1 = step(videoSrc1);
        frame = undistortImage(frame, cm1,'nearest');
        frame1 = undistortImage(frame1, cm2,'nearest');
        foreground = step(foregroundDetector, frame);
        foreground1 = step(foregroundDetector, frame1);
        ambl_foreground1 = zeros(size(foreground,1),...
            size(foreground,2));
        ambl_foreground2 = zeros(size(foreground1,1),...
            size(foreground1,2));
        for i = pos(1):pos(1)+pos(3)
            for j = pos(2):pos(2)+pos(4)
                ambl_foreground1(j,i) = foreground(j,i);
                ambl_foreground2(j,i) = foreground1(j,i);
            end
        end
        deteksi_objek(ambl_foreground1);
        deteksi_objek1(ambl_foreground2);
        filteredForeground = foreground;
        filteredForeground1 = foreground1;
        bbox = step(blobAnalysis, filteredForeground);
        bbox1 = step(blobAnalysis, filteredForeground1);
        if size(bbox,1) ~= 0 && size(bbox1,1) ~= 0
            jarak = focal_length/abs(bbox(1)-bbox1(1));
            numCars = jarak;
            result = insertObjectAnnotation(frame, 'rectangle', ...
                bbox,numCars,'Color', 'green');
            foreground = uint8(repmat(foreground, [1, 1, 3])) .* 255;
            foreground = insertObjectAnnotation(foreground,...
                'rectangle', bbox,numCars,'Color', 'red');
            axes(handles.axes4);
        end
    end
end

```

```

        imshow(result)
    %     numCars = numCars/100;
    tampil = strcat(num2str(numCars),' cm');
    set(handles.text5,'String',tampil);
    axes(handles.axes6);
    imshow(foreground)
end
clear bbox bbox1
end
%%

function deteksi_objek(layar)
    global foreground

    %     se = strel('square', 3);
    %     sa = strel('disk', 50);
    se = fspecial('gaussian',33);
    mask = layar;
    %
    % %     mask = imfilter(mask,se);
    % %     mask = imerode(mask,strel('rectangle', [17,17]));
    % %     mask = imdilate(mask,strel('rectangle', [17,17]));
    % % %     mask = imopen(mask, strel('rectangle', [8,8]));
    % %     mask = imclose(mask, strel('rectangle', [20, 20]));
    % %
    % %     mask = imfill(mask,'holes');
    %     mask = imopen(mask, se);
    %     mask = imclose(mask, sa);
    %     mask = imfill(mask,'holes');

    mask = imfilter(mask,se);
    mask = imerode(mask,strel('rectangle', [23,23]));
    %     mask = imdilate(mask,strel('rectangle', [17,17]));
    %     mask = imopen(mask, strel('rectangle', [8,8]));
    %     mask = imclose(mask, strel('rectangle', [20, 20]));

    mask = imfill(mask,'holes');
    foreground = logical(mask);
    %%

function deteksi_objek1(layar)
    global foreground1

    %     se = strel('square', 3);

```

```

%      sa = strel('disk', 50);
%      se = fspecial('gaussian',33);
%      mask = layar;

%
% %      mask = imfilter(mask,se);
% %      mask = imerode(mask,strel('rectangle', [17,17]));
% %      mask = imdilate(mask,strel('rectangle', [17,17]));
% % %      mask = imopen(mask, strel('rectangle', [8,8]));
% %      mask = imclose(mask, strel('rectangle', [20, 20]));
% %
% %      mask = imfill(mask,'holes');
%      mask = imopen(mask, se);
%      mask = imclose(mask, sa);
%      mask = imfill(mask,'holes');

%
%      mask = imfilter(mask,se);
%      mask = imerode(mask,strel('rectangle', [23,23]));
%      mask = imdilate(mask,strel('rectangle', [17,17]));
%      mask = imopen(mask, strel('rectangle', [8,8]));
%      mask = imclose(mask, strel('rectangle', [20, 20]));

%
%      mask = imfill(mask,'holes');
%      foreground1 = logical(mask);

```


BIODATA PENULIS



Penulis bernama Shofwan Ali Fauji, lahir di Gresik, 4 maret 1990. Penulis merupakan anak ketiga dari pasangan Nasrul dan Siti Munawaroh. Penulis menempuh pendidikan formal di SDN 1 Wedoroanom, SMPN 1 Kedamean dan SMAN 1 Driyorejo. Setelah lulus dari SMA penulis melanjutkan studi ke Jurusan Matematika UIN Maulana Malik Ibrahim Malang melalui jalur SNMPTN dan diterima sebagai mahasiswa angkatan 2008, dengan NIM 08610081 dan wisuda pada bulan Mei 2012 dengan judul skripsi **"ANALISIS FUNGSI AKTIVASI JARINGAN SYARAF TIRUAN UNTUK MENDETEKSI**

KARAKTERISTIK BENTUK GELOMBANG SPEKTRA BABI DAN SAPI". Kemudian penulis melanjutkan S2 di ITS pada tahun 2012 dengan NRP 1212201005. Penulis dapat dihubungi melalui e-mail: shofwan.ali.fauji12@mhs.its.ac.id